ISSN: 2321-2152

IJMECE

International Journal of modern electronics and communication engineering

E-Mail
editor.ijmece@gmail.com
editor@ijmece.com

www.ijmece.com

ISSN2321-2152 www.ijmece

Vol 8, Issuse.4 Oct 2020

Block chain Based Secure Voting System Using Iot

B.Venkateswarlu,[1]P.V.Sarath Chand[2],Raju Anthotil[3],E.Muralidhar Reddy[4],

ABSTRACT

A credible election is a critical mission for the survival of democracy, and the voter's trust is fundamental to civil stability. Too often, security, accessibility, verifiability, and transparency of electoral infrastructure have to remain untrustworthy. Preliminary reviews and surveys are made to related e-voting technologies, and comparative analysis conducted on the various authentication techniques used in voting frameworks. Recent security gains of blockchain technology are worthwhile going by the surge of crypto currencies, and as a suite of related protocols, it is distributed, irreversible, anonymous, incontrovertible, open source public asset ledger, where the complete data shared among all participants in the network. This paper investigates the development of an innovative voting model relying upon the use of blockchain and eye technology for verification and authentication of voters, and security of ballots in an anonymous voting scenario. The paper compares different eye technologies for identification and authentication of voters. A proposal is made for a multi-modal approach of both behavioural and physiological eyes techniques and includes a challenge response built within a blockchain voting platform to recognise voters. It highlights the components and prevailing security threats along with a review for the remedial approach. This aims to boost confidence and integrity of national and institutional democracy. Keywords: security; blockchain; smart contract; eye; e-voting; election; electronic democracy

INTRODUCTION

1.1 INTRODUCTION OF BLOCKCHAIN BASED SECURE VOTING SYSTEM USING IOT

In the world of democracy peoples have the fullest rights to decide and select an efficient leader to lead them. That decision is finalized by the process of election. The election is done by voting for the candidates who have opted to join in the election. The person who gets the most votes will win the election and he is decided to be the leader by the people, like Abraham Lincoln said "of the people, by the people, for the people". To maintain the integrity nowadays election commission is formed. In the starting, the election is done by voting to a candidate by the paper and then many vote rigging has been done in this method. But a same person has voted for more than one time and day by day this was increased dramatically in the election process. To avoid this, election commission has made a great change in the voting process by introducing an E-Voting machine. This machine consists of multiple buttons and each of the buttons is allocated for the separate candidate's symbol. When the voters press the button for a candidate, the vote count will be saved in the voting machine. It consists of a memory storage module which stores the data and finally the votes are counted by adding the votes in the machines storage unit. But also in this process, some peoples have done vote rigging by attempting to vote more than one time, and also by hacking the storage module in the voting machine. To over through all this types of problems we are proposing a system with more security, easy for voting and vote counting which set free time and money for next generation voting process. In this system block chain technology is implemented to provide security against the modifying the voting count through hacking, and the fingerprint module is used to evade the multiple appearances of the voters

Professor[1,2,3,4,] Assistant Professor[1,2,3,4], ,Associate Professor[1,2,3,4],
Department of CSE Engineering,
Pallavi Engineering College,
Mail ID:bvenkat1109@gmail.com, Mail ID:chandsarath70@gmail.com,
Kuntloor(V),Hayathnagar(M),Hyderabad,R.R.Dist.-501505

1.2 INTRODUCTION OF DOMAIN

Machine learning (**ML**)
Machine learning is the scientific study of algorithms and statistical models that computer systems use to perform a specific task without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of artificial intelligence. Machine learning algorithms build a mathematical model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to perform the task. Machine learning algorithms are used in a wide variety of applications, such as email filtering and computer vision, where it is difficult or infeasible to develop a conventional algorithm for effectively performing the task.

Machine learning is closely related to computational statistics, which focuses on making predictions using computers. The study of mathematical optimization delivers methods, theory and application domains to the field of machine learning. Data mining is a field of study within machine learning, and focuses on exploratory data analysis through unsupervised learning. In its application across business problems, machine learning is also referred to as predictive analytics.
The name machine learning was coined in 1959 by Arthur Samuel. Tom M. Mitchell provided a widely quoted, more formal definition of the algorithms studied in the machine learning field: "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E. This definition of the tasks in which machine learning is concerned offers a fundamentally operational definition rather than defining the field in cognitive terms. This follows Alan Turing's proposal in his paper "Computing Machinery and Intelligence, in which the question Can machines think?" is replaced with the question "Can machines do what we (as thinking entities) can do In Turing's proposal the various characteristics that could be possessed by a thinking machine and the various implications in constructing one are exposed.
Machine learning uses data to detect various patterns in a given dataset.
1.It can learn from past data and improve automatically.
2.It is a data-driven technology.
3.Machine learning is much similar to data mining as it also deals with the huge amount of the data.

How does Machine Learning Work?
A Machine Learning system learns from historical data, builds the prediction models, and whenever it receives new data, predicts the output for it. The accuracy of predicted output depends upon the amount of data, as the huge amount of data helps to build a better model which predicts the output more accurately.
Machine learning tasks are classified into several broad categories. In supervised learning, the algorithm builds a mathematical model from a set of data that contains both the inputs and the desired outputs. For example, if the task were determining whether an image contained a certain object, the training data for a supervised learning algorithm would include images with and without that object (the input), and each image would have a label (the output) designating whether it contained the object. In special cases, the input may be only partially available, or restricted to special feedback Semi-supervised learning algorithms develop mathematical models from incomplete training data, where a portion of the sample input doesn't have labels.
Classification algorithms
and regression algorithms are types of supervised learning. Classification algorithms are used when the outputs are restricted to a limited set of values. For a classification algorithm that filters emails, the input would be an incoming email, and the

output would be the name of the folder in which to file the email. For an algorithm that identifies spam emails, the output would be the prediction of either "spam" or "not spam", represented by the Boolean values true and false. Regression algorithms are named for their continuous outputs, meaning they may have any value within a range. Examples of a continuous value are the temperature, length, or price of an object.

In unsupervised learning, the algorithm builds a mathematical model from a set of data that contains only inputs and no desired output labels. Unsupervised learning algorithms are used to find structure in the data, like grouping or clustering of data points. Unsupervised learning can discover patterns in the data, and can group the inputs into categories, as in feature learning. Dimensionality reduction is the process of reducing the number of features, or inputs, in a set of data.

Active learning algorithms access the desired outputs (training labels) for a limited set of inputs based on a budget and optimize the choice of inputs for which it will acquire training labels. When used interactively, these can be presented to a human user for labeling. Reinforcement learning algorithms are given feedback in the form of positive or negative reinforcement in a dynamic environment and are used in autonomous vehicles or in learning to play a game against a human opponent Other specialized algorithms in machine learning include topic modeling, where the computer program is given a set of natural language documents and finds other documents that cover similar topics. Machine learning algorithms can be used to find the unobservable probability density function in density estimation problems. Meta learning algorithms learn their own inductive bias based on previous experience. In developmental robotics, robot learning algorithms generate their own sequences of learning experiences, also known as a curriculum, to cumulatively acquire new skills through self-guided exploration and social interaction with humans. These robots use guidance mechanisms such as active learning, maturation, motor synergies, and imitation.

Relation to data mining

Machine learning and data mining often employ the same methods and overlap significantly, but while machine learning focuses on prediction, based on known properties learned from the training data, data mining focuses on the discovery of (previously) unknown properties in the data (this is the analysis step of knowledge discovery in databases). Data mining uses many machine learning methods, but with different goals; on the other hand, machine learning also employs data mining methods as unsupervised learning or as a preprocessing step to improve learner accuracy. Much of the confusion between these two research communities (which do often have separate conferences and separate journals, ECML PKDD being a major exception) comes from the basic assumptions they work with: in machine learning, performance is usually evaluated with respect to the ability to reproduce known knowledge, while in knowledge discovery and data mining (KDD) the key task is the discovery of previously unknown knowledge. Evaluated with respect to known knowledge, an uninformed (unsupervised) method will easily be outperformed by other supervised
methods, while in a typical KDD task supervised methods cannot be used due to the unavailability of training data.

Relation to statistics

Machine learning and statistics are closely related fields in terms of methods, but distinct in their principal goal: statistics draws population inferences from a sample, while machine learning finds generalizable predictive patterns. According to Michael I. Jordan, the ideas of machine learning, from methodological principles to theoretical tools, have had a long pre-history in statistics. He also suggested the term data science as a placeholder to call the overall field.

Leo-Breiman distinguished two statistical modeling paradigms: data model and algorithmic model, wherein "algorithmic model" means more or less the machine learning algorithms like Random forest.

Some statisticians have adopted methods from machine learning, leading to a combined field that they call statistical learning.

Types of learning algorithms

The types of machine learning algorithms differ in their approach, the type of data they input and output, and the type of task or problem that they are intended to solve.

Supervised learning

Unsupervised learning

Reinforcement learning

1.3 INTRODUCTION TO THE DEEP LEARNING

Deep learning

Deep learning is a class of machine learning algorithms that uses multiple layers to progressively extract higher level features from the raw input. For example, in image processing, lower layers may identify edges, while higher layers may identify the concepts relevant to a human such as digits or letters or faces.

Deep learning (also known as deep structured learning or differential programming) is part of a broader family of machine learning methods based on artificial neural networks with representation learning. Learning can be supervised, semi-supervised or unsupervised.

Deep learning architectures such as deep neural networks, deep belief networks, recurrent neural networks and convolutional neural networks have been applied to fields including computer vision, speech recognition, natural language processing, audio recognition, social network filtering, machine translation, bioinformatics, drug design, medical image analysis, material inspection and board game programs, where they have produced results comparable to and in some cases surpassing human expert performance.

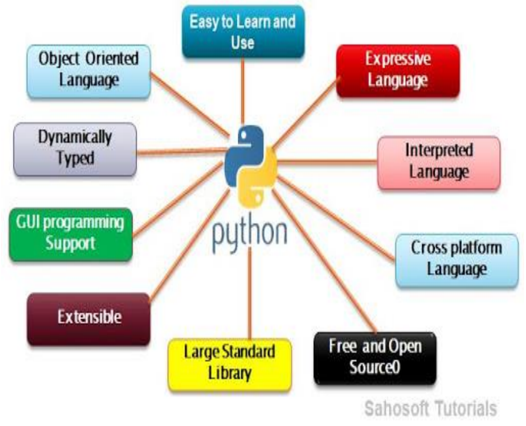
Artificial neural networks (ANNs) were inspired by information processing and distributed communication nodes in biological systems. ANNs have various differences from biological brains. Specifically, neural networks tend to be static and symbolic, while the biological brain of most living organisms is dynamic (plastic) and analog.

Deep learning is a subset of machine learning in artificial intelligence (AI) that has networks capable of learning unsupervised from data that is unstructured or unlabeled. Also known as deep neural learning or deep neural network.

2. LITRATURE STUDY

2.1 FEATURES OF PYTHON PROGRAMMING LANGUAGES

. **Python** is a dynamic, high level, free open source and interpreted programming language. It supports object-oriented programming as well as procedural oriented programming. In Python, we don't need to declare the type of variable because it is a dynamically typed language.
For example, x = 10
Here, x can be anything such as String, int, etc.



1.Easy to code:
Python is a high-level programming language. Python is very easy to learn the language as compared to other languages like C, C#, Javascript, Java, etc. It is very easy to code in python language and anybody can learn python basics in a few hours or days. It is also a developer-friendly language.

2. Free and Open Source:
Python language is freely available at the official website and you can download it from the given download link below click on the **Download Python** keyword.

3.Object-OrientedLanguage:
One of the key features of python is Object-Oriented programming. Python supports object-oriented language and concepts of classes, objects encapsulation, etc.
4. GUI Programming Support:
Graphical User interfaces can be made using a module such as PyQt5, PyQt4, wxPython, or Tk in python.

PyQt5 is the most popular option for creating graphical apps with Python.

5. High-Level Language: Python is a high-level language. When we write programs in python, we do not need to remember the system architecture, nor do we need to manage the memory.

6. Extensible feature: Python is a **Extensible** language. We can write us some Python code into C or C++ language and also we can compile that code in C/C++ language.

7. Python is Portable language: Python language is also a portable language. For example, if we have python code for windows and if we want to run this code on other platforms such as Linux, Unix, and Mac then we do not need to change it, we can run this code on any platform.

8. Python is Integrated language: Python is also an Integrated language because we can easily integrated python with other languages like c, c++, etc.

9. Interpreted Language: Python is an Interpreted Language because Python code is executed line by line at a time. like other languages C, C++, Java, etc. there is no need to compile python code this makes it easier to debug our code. The source code of python is converted into an immediate form called **bytecode**.

10. Large Standard Library Python has a large standard library which provides a rich set of module and functions so you do not have to write your own code for every single thing. There are many libraries present in python for such as regular expressions, unit-testing, web browsers, etc.

11. Dynamically Typed Language: Python is a dynamically-typed language. That means the type (for example- int, double, long, etc.) for a variable is decided at run time not in advance because of this feature we don't need to specify the type of variable.

Python is used for:

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

Python do?:

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

Why Python?:

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

Python compared to other programming languages

- Python was designed for readability, and has some similarities to the English language with influence from mathematics.
- Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.

2.2 SOFTWARE ENVIRONMENT:.

Python installation procedure:

Windows Based

It is highly unlikely that your Windows system shipped with Python already installed. Windows systems typically do not. Fortunately, installing does not involve much more than downloading the Python installer from the python.org website and running it. Let's take a look at how to install Python 3 on Windows:

Step 1: Download the Python 3 Installer

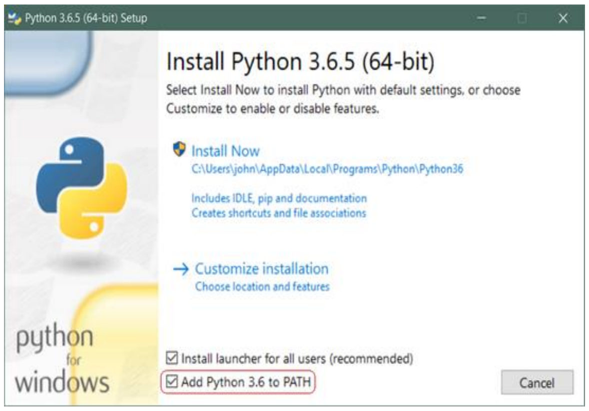
1. Open a browser window and navigate to the Download page for Windows at python.org.
2. Underneath the heading at the top that says **Python Releases for Windows**, click on the link for the **Latest Python 3 Release - Python 3.x.x**. (As of this writing, the latest is Python 3.6.5.)
3. Scroll to the bottom and select either **Windows x86-64 executable installer** for 64-bit or **Windows x86 executable installer** for 32-bit. (See below.)

Sidebar: 32-bit or 64-bit Python?
For Windows, you can choose either the 32-bit or 64-bit installer. Here's what the difference between the two comes down to:

- If your system has a 32-bit processor, then you should choose the 32-bit installer.
- On a 64-bit system, either installer will actually work for most purposes. The 32-bit version will generally use less memory, but the 64-bit version performs better for applications with intensive computation.
- If you're unsure which version to pick, go with the 64-bit version.

Note: Remember that if you get this choice "wrong" and would like to switch to another version of Python, you can just uninstall Python and then re-install it by downloading another installer from python.org.

Step 2: Run the Installer
Once you have chosen and downloaded an installer, simply run it by double-clicking on the downloaded file. A dialog should appear that looks something like this:


Python 3.6.5 (64-bit) Setup

Install Python 3.6.5 (64-bit)

Select Install Now to install Python with default settings, or choose Customize to enable or disable features.

Install Now
C:\Users\john\AppData\Local\Programs\Python\Python36

Includes IDLE, pip and documentation
Creates shortcuts and file associations

→ Customize installation
Choose location and features

☑ Install launcher for all users (recommended)
☑ Add Python 3.6 to PATH

Cancel

Important: You want to be sure to check the box that says **Add Python 3.x to PATH** as shown to ensure that the interpreter will be placed in your execution path.
Then just click **Install Now**. That should be all there is to it. A few minutes later you should have a working Python 3 installation on your system.

Mac OS based

While current versions of macOS (previously known as "Mac OS X") include a version of Python 2, it is likely out of date by a few months. Also, this tutorial series uses Python 3, so let's get you upgraded to that.
The best way we found to install Python 3 on macOS is through the Homebrew package manager. This approach is also recommended by community guides like The Hitchhiker's Guide to Python.

Step 1: Install Homebrew (Part 1)
To get started, you first want to install Homebrew:

1. Open a browser and navigate to http://brew.sh/. After the page has finished loading, **select the Homebrew bootstrap code under "Install Homebrew"**. Then hit cmd+c to copy it to the clipboard. Make sure you've captured the text of the complete command because otherwise the installation will fail.
2. Now you need to **open a Terminal app window, paste the Homebrew bootstrap code, and then hit** Enter. This will begin the Homebrew installation.
3. If you're doing this on a fresh install of macOS, you may get a pop up alert **asking you to install Apple's "command line developer tools"**. You'll

need those to continue with the installation, so please **confirm the dialog box by clicking on "Install"**.

At this point, you're likely waiting for the command line developer tools to finish installing, and that's going to take a few minutes. Time to grab a coffee or tea!

Step 2: Install Homebrew (Part 2)

You can continue installing Homebrew and then Python after the command line developer tools installation is complete:

1. Confirm the "The software was installed" dialog from the developer tools installer.
2. Back in the terminal, hit Enter to continue with the Homebrew installation.
3. Homebrew asks you to enter your password so it can finalize the installation. **Enter your user account password and hit** Enter to continue.
4. Depending on your internet connection, Homebrew will take a few minutes to download its required files. Once the installation is complete, you'll end up back at the command prompt in your terminal window.

Whew! Now that the Homebrew package manager is set up, let's continue on with installing Python 3 on your system.

Step 3: Install Python

Once Homebrew has finished installing, **return to your terminal and run the following command**:

```
$ brew install python3
```

Note: When you copy this command, be sure you don't include the $ character at the beginning. That's just an indicator that this is a console command.

This will download and install the latest version of Python. After the Homebrew brew install command finishes, Python 3 should be installed on your system.

You can make sure everything went correctly by testing if Python can be accessed from the terminal:

1. Open the terminal by launching **Terminal app**.
2. Type pip3 and hit Enter.
3. You should see the help text from Python's "Pip" package manager. If you get an error message running pip3, go through the Python install steps again to make sure you have a working Python installation.

Assuming everything went well and you saw the output from Pip in your command prompt window…congratulations! You just installed Python on your system, and you're all set to continue with the next section in this tutorial.

Packages need for python based programming:

- **Numpy**
  NumPy is a Python package which stands for 'Numerical Python'. It is the core library for scientific computing, which contains a powerful n-dimensional array object, provide tools for integrating C, C++ etc. It is also useful in linear algebra, random number capability etc.
- **Pandas**
  Pandas is a high-level data manipulation tool developed by Wes McKinney. It is built on the Numpy package and its key data structure is called the DataFrame. DataFrames allow you to store and manipulate tabular data in rows of observations and columns of variables.
- **Keras**
  Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano. Use Keras if you need a deep learning library that: Allows for easy and fast prototyping (through user friendliness, modularity, and extensibility).
- **Sklearn**
  Scikit-learn is a free machine learning library for Python. It features various algorithms like support vector machine, random forests, and k-neighbours, and it also supports Python numerical and scientific libraries like NumPy and SciPy.
- **Scipy**
  SciPy is an open-source Python library which is used to solve scientific and mathematical problems. It is built on the NumPy extension and allows the user to manipulate and visualize data with a wide range of high-level commands.
- **Tensorflow**

TensorFlow is a Python library for fast numerical computing created and released by Google. It is a foundation library that can be used to create Deep Learning models directly or by using wrapper libraries that simplify the process built on top of TensorFlow.

- **Django**
  Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source.
- **Pyodbc**
  pyodbc is an open source Python module that makes accessing ODBC databases simple. It implements the DB API 2.0 specification but is packed with even more Pythonic convenience. Precompiled binary wheels are provided for most Python versions on Windows and macOS. On other operating systems this will build from source.
- **Matplotlib**
  Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack. It was introduced by John Hunter in the year 2002.
- **Opencv**
  OpenCV-Python is a library of Python bindings designed to solve computer vision problems. Python is a general purpose programming language started by Guido van Rossum that became very popular very quickly, mainly because of its simplicity and code readability.
- **Nltk**
  Natural Language Processing with Python NLTK is one of the leading platforms for working with human language data and Python, the module NLTK is used for natural language processing. NLTK is literally an acronym for Natural Language Toolkit. In this article you will learn how to tokenize data (by words and sentences).
- **SQLAIchemy**
  SQLAlchemy is a library that facilitates the communication between Python programs and databases. Most of the times, this library is used as an Object Relational Mapper (ORM) tool that translates Python classes to tables on relational databases and automatically converts function calls to SQL statements.
- **Urllib**
  urllib is a Python module that can be used for opening URLs. It defines functions and classes to help in URL actions. With Python you can also access and retrieve data from the internet like XML, HTML, JSON, etc. You can also use Python to work with this data directly.

Installation of packages:

Syntax for installation of packages via cmd terminal using the basic

Step:1- First check pip cmd

First check pip cmd

If ok then

Step:2- pip list

Check the list of packages installed and then install required by following cmds

Step:3- pip install package name

The package name should as requirement

2.3 OPENCV INSTALLATION:

Open cv:

OpenCV was started at Intel in 1999 by Gary Bradsky and the first release came out in 2000. Vadim Pisarevsky joined Gary Bradsky to manage Intel's Russian software OpenCV team. In 2005, OpenCV was used on Stanley, the vehicle who won 2005 DARPA Grand Challenge. Later its active development continued under the support of Willow Garage, with Gary Bradsky and Vadim Pisarevsky leading the project. Right now, OpenCV supports a lot of algorithms related to Computer Vision and Machine Learning and it is expanding day-by-day. Currently OpenCV supports a wide variety of programming languages like C++, Python, Java etc and is available on different platforms including Windows, Linux, OS X, Android, iOS etc. Also, interfaces based on CUDA and OpenCL are also under active development for high-speed GPU operations. OpenCV-Python is the Python API of OpenCV. It combines the best qualities of OpenCV C++ API and Python language.

OpenCV-Python Python is a general purpose programming language started by Guido van Rossum, which became very popular in short time mainly because of its simplicity and code readability. It enables the programmer to express his ideas in fewer lines of code without reducing any readability. Compared to other languages like C/C++, Python is slower. But another important feature of Python is that it can be easily extended with C/C++. This feature helps us to write computationally intensive codes in C/C++ and create a Python wrapper for it so that we can use these wrappers as Python modules. This gives us two advantages: first, our code is as fast as original C/C++ code (since it is the actual C++ code working in background) and second, it is very easy to code in Python. This is how OpenCV-Python works, it is a Python wrapper around original C++ implementation. And the support of Numpy makes the task more easier. Numpy is a highly optimized library for numerical operations. It gives MATLAB-style syntax. All the OpenCV array structures are converted to-and-from Numpy arrays. So whatever operations you can do in Numpy, you can combine it with OpenCV, which increases number of weapons in your arsenal. Besides that, several other libraries like SciPy,

Matplotlib which supports Numpy can be used with this. So OpenCV-Python is an appropriate tool for fast prototyping of computer vision problems.

Since OpenCV is an open source initiative, all are welcome to make contributions to this library. And it is same for this tutorial also. So, if you find any mistake in this tutorial (whether it be a small spelling mistake or a big error in code or concepts, whatever), feel free to correct it. 1.1. Introduction to OpenCV 7 OpenCV-Python Tutorials Documentation, Release 1 And that will be a good task for fresher's who begin to contribute to open source projects. Just fork the OpenCV in github, make necessary corrections and send a pull request to OpenCV.

OpenCV developers will check your pull request, give you important feedback and once it passes the approval of the reviewer, it will be merged to OpenCV. Then you become a open source contributor. Similar is the case with other tutorials, documentation etc. As new modules are added to OpenCV-Python, this tutorial will have to be expanded. So those who knows about particular algorithm can write up a tutorial which includes a basic theory of the algorithm and a code showing basic usage of the algorithm and submit it to OpenCV. Remember, we together can make this project a great success!!! Contributors Below is the list of contributors who submitted tutorials to OpenCV-Python.

1. Alexander Mordvintsev (GSoC-2013 mentor)

2. Abid Rahman K. (GSoC-2013 intern)

Additional Resources

1. A Quick guide to Python - A Byte of Python

2. Basic Numpy Tutorials

3. Numpy Examples List

4. OpenCV Documentation

5. OpenCV Forum

Install OpenCV-Python in Windows

Goals In this tutorial

We will learn to setup OpenCV-Python in your Windows system. Below steps are tested in a Windows 7-64 bit machine with Visual Studio 2010 and Visual Studio 2012. The screenshots shows VS2012.

Installing Open CV from prebuilt binaries

1. Below Python packages are to be downloaded and installed to their default locations.

1.1. Python-2.7.x.

1.2. Numpy.

1.3. Matplotlib (Matplotlib is optional, but recommended since we use it a lot in our tutorials).

2. Install all packages into their default locations. Python will be installed to C:/Python27/.

3. After installation, open Python IDLE. Enter import numpy and make sure Numpy is working fine.

4. Download latest OpenCV release from source forge site and double-click to extract it.

5. Goto opencv/build/python/2.7 folder.

6. Copy cv2.pyd to C:/Python27/lib/site-packeges.

7. Open Python IDLE and type following codes in Python terminal.

```
>>> import cv2

>>> print cv2.__version__
```

If the results are printed out without any errors, congratulations!!! You have installed OpenCV-Python successful

Download and install necessary Python packages to their default locations

1. Python 3.6.8.x

2. Numpy

3. Matplotlib (Matplotlib is optional, but recommended since we use it a lot in our tutorials.)

Make sure Python and Numpy are working fine.

4. Download OpenCV source. It can be from Source forge (for official release version) or from Github (for latest source).

5. Extract it to a folder, opencv and create a new folder build in it.

6. Open CMake-gui (Start > All Programs > CMake-gui)

7. Fill the fields as follows (see the image below):

7.1. Click on Browse Source... and locate the opencv folder.

7.2. Click on Browse Build... and locate the build folder we created.

7.3. Click on Configure.

7.4. It will open a new window to select the compiler. Choose appropriate compiler (here, Visual Studio 11) and click Finish.

7.5. Wait until analysis is finished.

8. You will see all the fields are marked in red. Click on the WITH field to expand it. It decides what extra features you need. So mark appropriate fields. See the below image:

9. Now click on BUILD field to expand it. First few fields configure the build method. See the below image:

10. Remaining fields specify what modules are to be built. Since GPU modules are not yet supported by Open CV Python, you can completely avoid it to save time (But if you work with them, keep it there). See the image below:

11. Now click on ENABLE field to expand it. Make sure ENABLE_SOLUTION_FOLDERS is unchecked (Solution folders are not supported by Visual Studio Express edition). See the image below:

12. Also make sure that in the PYTHON field, everything is filled. (Ignore PYTHON_DEBUG_LIBRARY). See image below:

13. Finally click the Generate button.

14. Now go to our opencv/build folder. There you will find OpenCV.sln file. Open it with Visual Studio.

15. Check build mode as Release instead of Debug.

16. In the solution explorer, right-click on the Solution (or ALL_BUILD) and build it. It will take some time to finish.

17. Again, right-click on INSTALL and build it. Now OpenCV-Python will be installed.

18. Open Python IDLE and enter import cv2. If no error, it is installed correctly

Using OpenCV Read an image

Use the function cv2.imread () to read an image. The image should be in the working directory or a full path of image should be given. Second argument is a flag which specifies the way image should be read.

cv2.IMREAD_COLOR : Loads a color image. Any transparency of image will be neglected. It is the default flag.

cv2.IMREAD_GRAYSCALE : Loads image in gray scale mode

cv2.IMREAD_UNCHANGED : Loads image as such including alpha channel

See the code below:

- import numpy as np
- import cv2
- # Load an color image in grayscale
- img = cv2.imread('messi5.jpg',0)

Warning: Even if the image path is wrong, it won't throw any error, but print img will give you None

Display an image Use the function cv2.imshow() to display an image in a window. The window automatically fits to the image size. First argument is a window name which is a string. second argument is our image. You can create as many windows as you wish, but with different window names.

- cv2.imshow('image', mg)
- cv2.waitKey(0)

cv2.waitKey() is a keyboard binding function. Its argument is the time in milliseconds.

- cv2.destroyAllWindows()

cv2.destroyAllWindows () simply destroys all the windows we created

Write an image

Use the function cv2.imwrite () to save an image. First argument is the file name, second argument is the image you want to save.

- cv2.imwrite('messigray.png',img)

This will save the image in PNG format in the working directory

Below program loads an image in gray scale, displays it, save the image if you press 's' and exit, or simply exit without saving if you press ESC key.

- import numpy as np
- import cv2
- img = cv2.imread('messi5.jpg',0)
- cv2.imshow('image', mg)
- k = cv2.waitKey(0)
- if k == 27:   # wait for ESC key to exit
- cv2.destroyAllWindows()
- elif k == ord('s'): # wait for 's' key to save and exit
- cv2.imwrite('messigray.png',img)
  - cv2.destroyAllWindows()

3.SYSTEM ANALYSIS

3.1.1 EXISTING SYSTEM

- ❖ The dataset used for this is real and authentic. The dataset is acquired from UCI machine learning repository website.
- ❖ The title of the dataset is 'Crime and Communities'. It is prepared using real data from socio-economic data from 1990 US Census, law enforcement data from the 1990 US LEMAS survey and crime data from the 1995 block chain technology UCR.
- ❖ This dataset contains a total number of 147 attributes and 2216 instances.

3.1.2 PROPOSED SYSTEM :

- ❖ From a large list of attributes, only eighteen attributes are chosen for Exploratory Data Analysis. The chosen attributes are namely state, HousVacant,PctHouseOccup, PctHouseOwnCC, PctVacantBoarded, PctVacMore6Mos, PctUnemployed, PctEmploy, murdperPop, rapesperPop, robbperPop, assaultperPop, burglperPop, larcperPop, autoTheftperPop, arsonsperPop, nonviolperpop and violentcrimesperpop.

- ❖ Regression Analysis is limited to the following predictor and response variable predictor variables: Housevacant, PctHouseOccup, PctHouseownCC, PctVacantBoarded, PctVacmore6Mos, PctUnemployed, PctEmploy Response variables: Violentcrimesperpop.

- ❖ Solving the imbalanced class problem, the machine learning agent was able to categorize crimes with 81% accuracy.

3.2 SYSTEM SPECIFICATION:

3.2.1 HARDWARE REQUIREMENTS:

- ❖ System : Pentium IV 2.4 GHz.
- ❖ Hard Disk : 40 GB.
- ❖ Floppy Drive : 1.44 Mb.
- ❖ Monitor : 14’ Colour Monitor.
- ❖ Mouse : Optical Mouse.
- ❖ Ram : 512 Mb.

3.2.2 SOFTWARE REQUIREMENTS:

- ❖ Operating system : Windows 7 Ultimate.
- ❖ Coding Language : Python.
- ❖ Front-End : Python.
- ❖ Designing : Html,css,javascript.
- ❖ Data Base : MySQL.

4.SYSTEM DESIGN

4.1 Introduction of voting system

The participation of the citizen in politics has been declining over the years due to loss of interest and confidence in the electoral process, all of which culminates from human error to flaws in security and integrity of the system (Stein and Wenda, 2014). The need to enhance the citizen’s confidence and add credibility to the voting models result in too many countries introducing electronic voting technology in their elections, and as (Christidis and Devetsiokiotis, 2016) suggests, is an attempt to boost the efficiency and integrity of the system. However, while implementing the use of electronic voting (e-voting) which is synonymous with the use of technology and electronic devices to conduct an election, it should be ensured that the system must protect voter privacy, and be tamper-resistant to various electoral fraud like ballot stuffing, incorrect tallying, etc., while still maintaining vote accuracy (Cunningham, Bernhard and Halderman, 2016).

In addition, while designing an e-voting machine, human factors should be considered in the sense that all range of possible voters can use it successfully (Bhatia and Gupta, 2014).

Blockchain has been recognised as a distributed computing framework wherein network nodes execute and keep the record of every transaction, grouped in blocks for subsequent reference (Clack, Bakshi and Braine, 2016). It is difficult to manipulate and compromise due to distributed consensus among participants and difficult to erase or be manipulated by fraud. To prevent identity falsification, Eyes use the human unique physiological or behavioural characteristics to identify or verify a person (Aroni and Maneai, 2014).

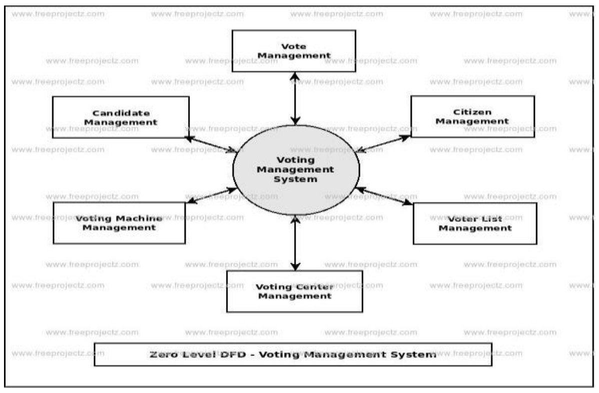
Voter's identity in a cryptographically based identification system of the blockchain offers genuine confidence for voter's eligibility and privacy. It promotes trust, anonymity, and immutability of ballots. Therefore, this research presents preliminary reviews and surveys related to e-voting technologies. Comparative analysis are conducted on the various authentication techniques used in e-voting systems. A presentation is made for tailored analysis of blockchain and eye techniques for the security of an e-voting system. Identification for security weaknesses is made for both techniques. A recommendation for bi-modal approach is proffered towards an improved secure e-voting system.requirement for establishing any kind of trust framework among parties who do not otherwise lacks trust (Weizhi, et al., 2018). Accordingly, smart contracts also give rise to the concept of Decentralised Autonomous Organisations (DAOs), a tool to create democratic organisations that are resilient to corruption (Richard, 2017).

The organisational bylaws are mathematically encoded such that they lack a bottleneck or single point of failure, while being autonomous, running themselves without human intervention. 2.3. Biometric Technology The smart Biometrics refer to the use of human unique physiological or behavioural characteristics to identify or verify a person (Aroni and Maneai, 2014). Biometrics measure individuals' unique physical or behavioural characteristics to recognise or authenticate their identity (Simon, and Mark, 2001).

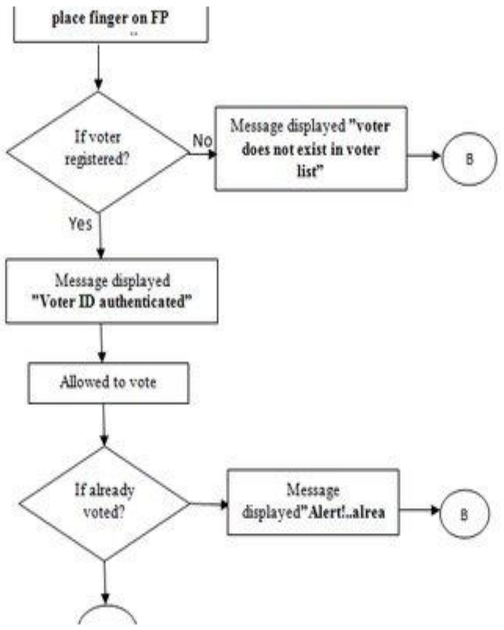
Related biometric authentication devices in (Wayman, 2009) aid to prove identity and use unique traits and behavioural features in the form of a fingerprint, iris, voice, and facial recognition to identify and authenticate access to electronic assets. Two distinct characteristics of biometrics are explained next. Physiological characteristics: These characteristics are physically present in humans, like the fingerprint, face, eye, iris, retina, and hand geometry (Aroni and Maneai, 2014).

Behavioural characteristics: These are characteristics that are generated by the action of an individual; the latter indirectly measures the characteristics of the individual body like signature, voice, and keystrokes.

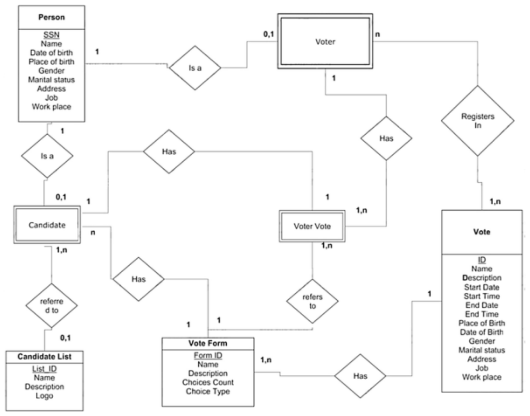
4.2 DATA FLOW DIAGRAM



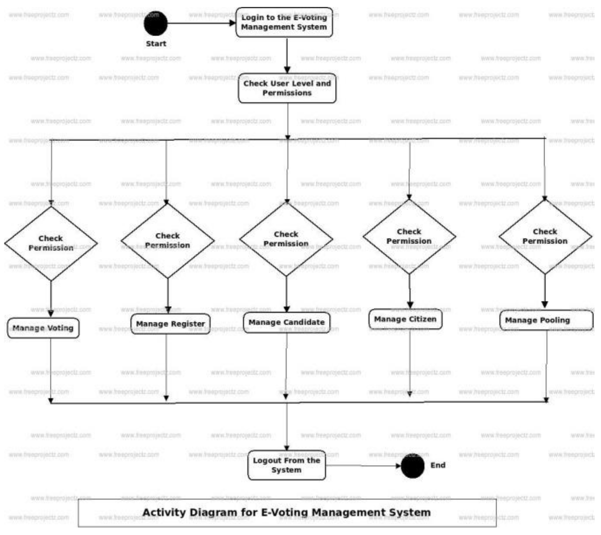
4.3 FLOW CHART


place finger on FP
If voter registered?
No
Message displayed "voter does not exist in voter list"
B
Yes
Message displayed "Voter ID authenticated"
Allowed to vote
If already voted?
Message displayed"Alert!..alrea
B

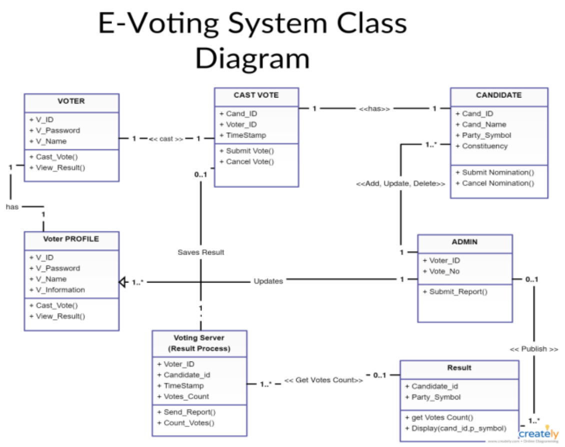
4.4 E-R DIAGRAM



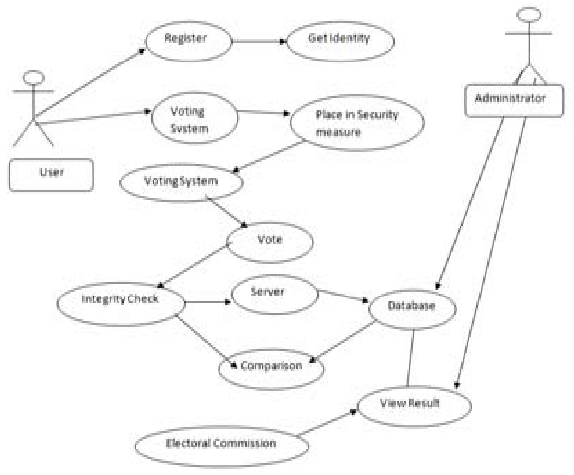
4.5 UML DIAGRAM



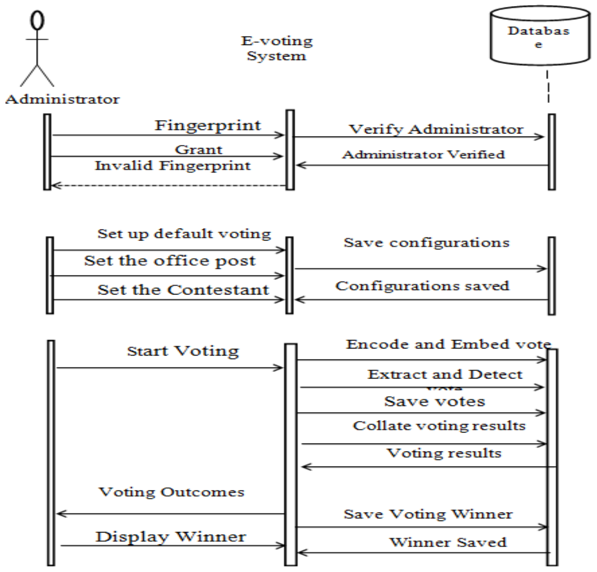
4.5.1 CLASS DIAGRAM



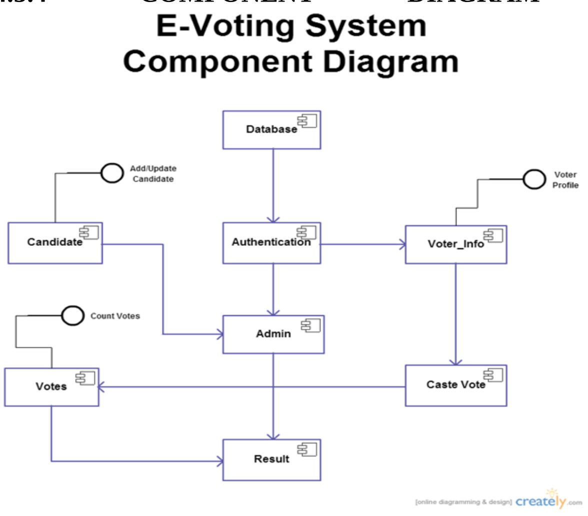
4.5.2 USE CASE DIAGRAM



4.5.3 SEQUENCE DIAGRAM



4.5.4 COMPONENT DIAGRAM



4.6 Input and Output Designs:

4.6.1 INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

OBJECTIVES

1.Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

2.It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

3.When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

4.6.2 OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

2.Select methods for presenting information.

3.Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

- Convey information about past activities, current status or projections of the
- Future.
- Signal important events, opportunities, problems, or warnings.
- Trigger an action.
- Confirm an action.

6. SYSTEM TESTING AND IMPLEMENTATION

6.1SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

TYPES OF TESTS :

Unit testing :

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing :
Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test : Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test :
System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing :
White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

6.2 Integration Testing :

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

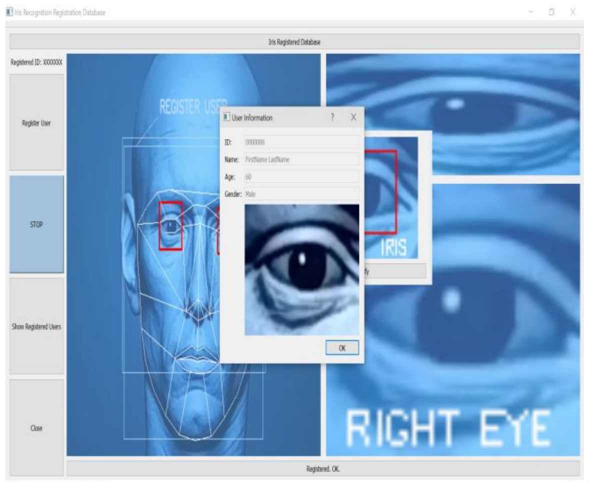
Test Results: All the test cases mentioned above passed successfully. No defects encountered.

6.3 Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

7. SCREENSHOTS



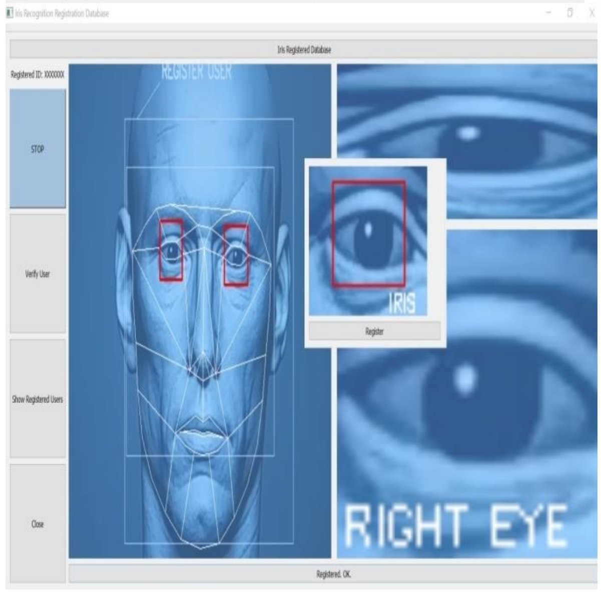
User Information

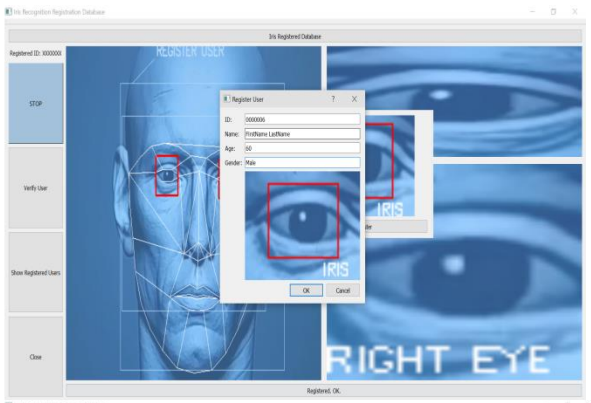
ID: 0000006

Name: FirstName LastName

Age: 60

Gender: Male





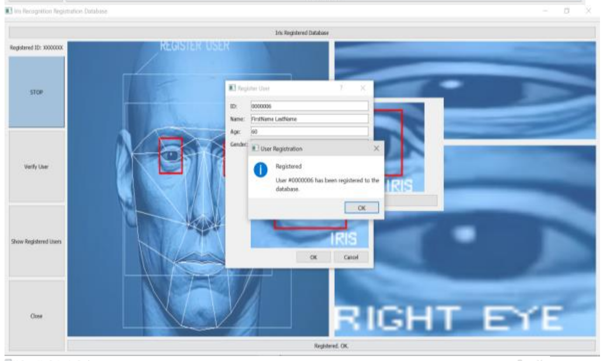
Register User

ID: 0000006

Name: FirstName LastName

Age: 60

Gender: Male



Register User

ID: 0000006

Name: FirstName LastName

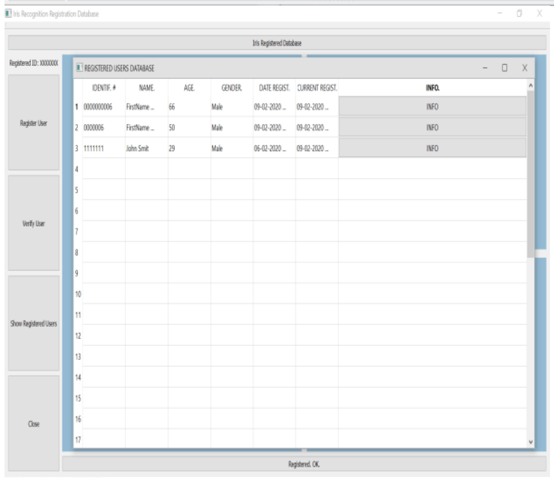
Age: 60

User Registration

Registered

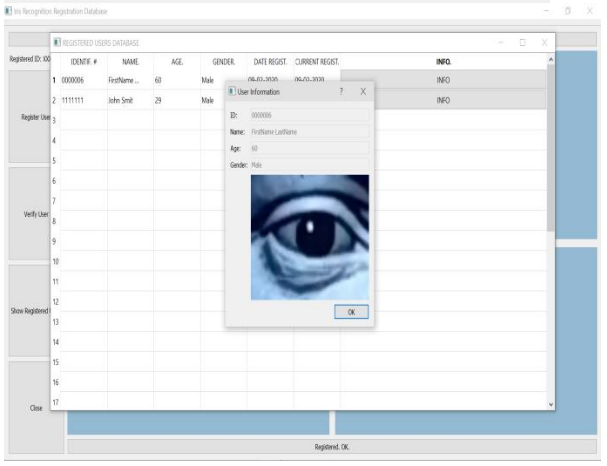
User #0000006 has been registered to the database.

OK



REGISTERED USERS DATABASE

	IDENTIF. #	NAME.	AGE.	GENDER.	DATE REGIST.	CURRENT REGIST.	INFO.
1	000000006	FirstName ...	66	Male	09-02-2020 ...	09-02-2020 ...	INFO
2	0000006	FirstName ...	50	Male	09-02-2020 ...	09-02-2020 ...	INFO
3	1111111	John Smit	29	Male	06-02-2020 ...	09-02-2020 ...	INFO
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							



REGISTERED USERS DATABASE

	IDENTIF. #	NAME.	AGE.	GENDER.	DATE REGIST.	CURRENT REGIST.	INFO.
1	0000006	FirstName ...	60	Male	09-02-2020	09-02-2020	INFO
2	1111111	John Smit	29	Male			INFO
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							

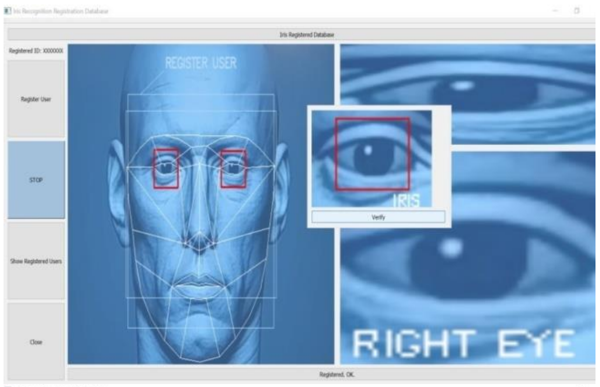
User Information

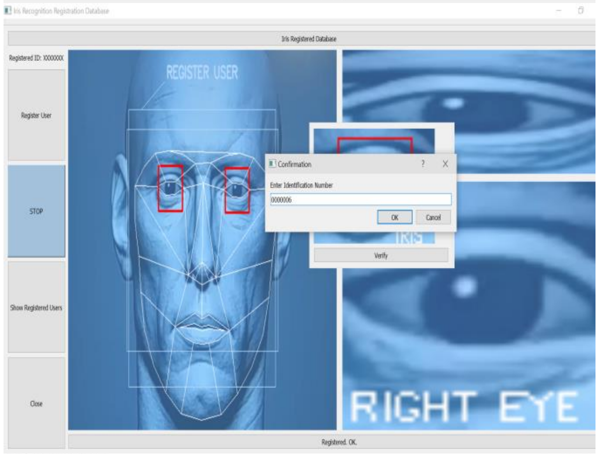
ID: 0000006

Name: FirstName LastName

Age: 60

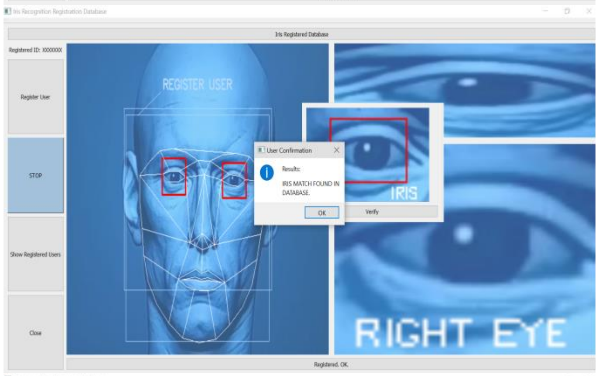
Gender: Male





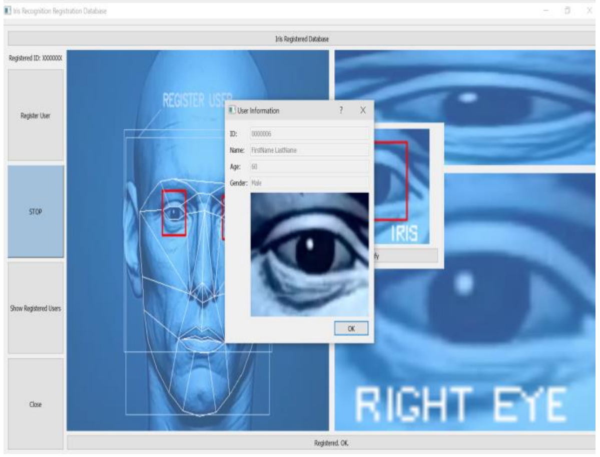
Enter Identification Number

0000006



Results:

IRIS MATCH FOUND IN DATABASE.

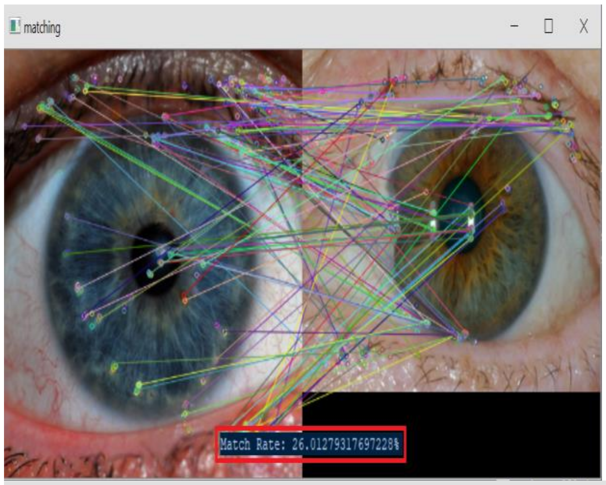


ID: 0000006

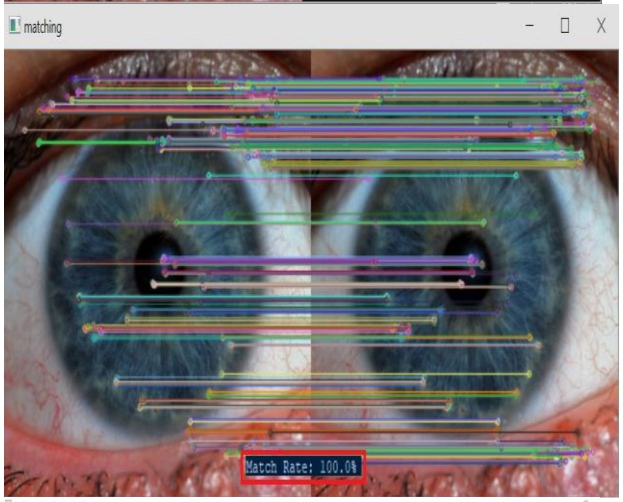
Name: FirstName LastName

Age: 60

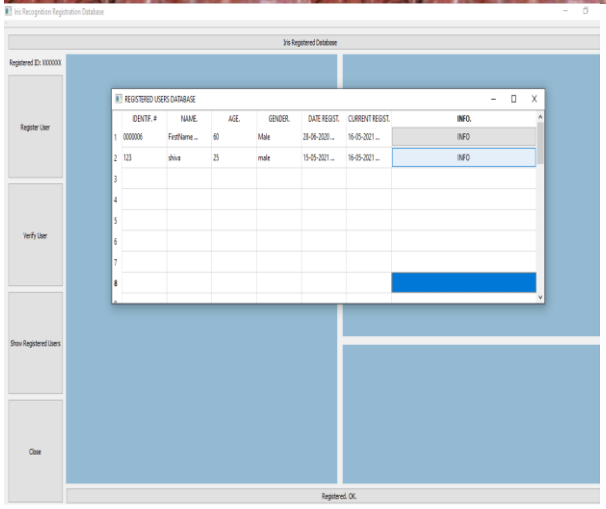
Gender: Male



Match Rate: 26.012793176972228%



Match Rate: 100.0%



	IDENTIF. #	NAME.	AGE.	GENDER.	DATE REGIST.	CURRENT REGIST.	INFO.
1	0000006	FirstName ...	60	Male	28-06-2020 ...	16-05-2021 ...	INFO
2	123	shiva	25	male	15-05-2021 ...	16-05-2021 ...	INFO
3							
4							
5							
6							
7							
8							

8. CONCLUSION

The paper offered a glimpse of challenges to electronic democracy and identified current biometric technologies and security anomalies of existing e-voting technologies. From the review of research findings, the study made first-hand presentation of multimodal biometric approach for identification and authentication of voters. The risk against future e-voting systems is conceivable, this prompted our proposal for bimodal integration of biometric and blockchain technologies for verifiable identity and vote audit trail that would anchor needed parameters for the secure voting system. Cyber resilient electoral infrastructure, openness and audit of voting information will likely increase public trust. It is therefore unnecessary for the e-voting system vendors to keep hiding relevant information to the public because people should be able to criticize and analyse the security and efficiency of the e-voting system as the election is the backbone of democracy, and to have confidence that election result reflects the true will of the people. Security and privacy gains of the blockchain technology are obvious and its adoption in an electronic democracy is hereby recommended. An integrated biometric identification and authentication techniques in a decentralised voting application if launched on blockchain network offer resilient security and confidence in the promotion of electronic democracy.

Continuous innovations in biometric technologies keep hope alive towards identity management solution to recognise, verify and authenticate eligible voters in the choice of democratic governance. Of course, Blockchain is promising with widespread adoption by previously sceptical innovators. The study does not present a die-cast security guarantee of the blockchain, and it is due to unpredicted disruptions in technology and limitations by design. It would require designing a decentralised infrastructure where every identity is controlled not by a trusted third-party, but by its principal owner. This is an impossibility now and developers would have to contend with evolving cyber threats and take measures towards a more secure e-voting model.

References

1. ***Agresti, A. and Presnell, B. (2000) 'Statistical Issues in the 2000 U.S. Presidential Election in Florida', University of Florida Journal of Law and Public Policy, 1st ed. (ebook) Florida. Available at: http://www.stat.ufl.edu/~aa/articles/agresti_presnell_2001.pdf (Accessed 12 Jan. 2017).***
2. ***Agwanshi, K. K and Dubey, S. (2012) 'Biometric Authentication using Human Footprint,' International Journal of Applied Information Systems (IJAIS) –ISSN: 2249-0868. Foundation of Computer Science FCS, New York, USAVolume 3–No.7, August2012. DOI: 10.5120/ijais12-450568***
3. ***Akanksha, A and Manoj, K. V. (2016) 'Multimodal Biometric Systems– A Survey', International Journal of Advanced Research in Computer Science and Software Engineering, Volume 6, Issue 3, March 2016. Available at: http://ijarcsse.com/Before_August_2017/docs/papers/Volume_6/3_March2016/V6I3-0320.pdf Alfred, C. (2015) 'Critical and Peculiar Challenges of the 2015 Nigeria's General Elections in the Core Niger Delta: A Participant Observer's Perspective', Independent National Electoral Commission (INEC) as a special feature that added credibility to the electoral system in the 2015 general election. Available at: http://www.inecnigeria.org/wp-content/uploads/2015/07/Conference-Paper-by-Charles-Alfred.pdf Alternative Voting Technologies Report (2013) 'Ontario: Chief Electoral Officer's Submission to the Legislative Assembly,' (online) http://www.elections.on.ca/content/dam/NGW/sitecontent/2014/reports/Alternative%20Voting%20Technologies%20Report%20(2012).pdf (Accessed 12 Jan. 2017)***
4. ***Ansper, A., Buldas, A., Oruaas, M., Priisalu, J., Veldre, A., Willemson, J and Virunurm, K. (2003) 'E-voting concept security: analysis and measures', Technical Report EH-02-01, Estonian National Electoral Committee. Aroni, I and Maneai, A. (2014) 'Recognition of a Person based on the Characteristics of the Iris and Retina', Bulletin of the Transilvania University of Braşov Series VII: Social Sciences, Law, Vol. 7 (56) No. 1 – 2014. Available at: http://webbut.unitbv.ro/BU2014/Series%20VII/BULETIN%20VII%20PDF/03_AROn-MANEA-1_2014.pdf Ashok, J., Shivashankar, V and Mudiraj, P. V. (2010) 'An Overview of Biometrics,' International Journal on Computer Science and Engineering. Vol. 02, No. 07, pp 2402-2408. Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.301.2569&rep=rep1&type=pdf Azougaghe,***
5. ***A., Kartit, Z., Hedabou, M., Belkasmi, M., Benmiloud, M and Marraki, M. (2015) 'An Efficient Electronic Voting System in a Cloud Computing Environment', International Review on Computers and Software (IRECOS), vol. 10, no. 11, 30 Nov. 2015, pp. 1103-09, doi:10.15866/irecos.v10i11.7667***
6. ***Barbara, S and Douglas, W. (2012) 'Internet Voting in the U.S.', Jones Communications of the ACM, Vol. 55 No. 10, Pages 68-77. doi:10.1145/2347736.2347754 Benson, J (2011) 'SOS - Voting Equipment,' (online) http://www.michigan.gov/sos/0,1607,7-127-1633_8716_45458---,00.html (Accessed 12 Jan. 2017).***
7. ***Bhatia, V., & Gupta, R. (2014) 'A novel electronic voting machine design with voter information facility using microcontroller'. 2014 International Conference on Computing for Sustainable Global Development (INDIACom), 274–276. doi: 10.1109/IndiaCom.2014.6828142 (Accessed 11 Jan. 2017).***
8. ***Biel, I., Pettersson, O., Philipson, L and Wide, P. (2001) 'ECG Analysis, A New Approach in Human Identification', IEEE Transactions on Instrumentation and Measurement, vol.50, no.3, pp. 808 – 812, June 2001. Biometric Technology***

Today (2006) ‘Mexico deploys multi-biometric voting system.’ ScienceDirect (Online service) Publisher: Elsevier Advanced Technology. ISSN: 0969-4765, vol. 14, no. 5, May. 2006, pp. 3-4, doi:10.1016/S0969-4765(06)70519-7

9. *Biometric Technology Today (2015) ‘Ghana, Tanzania and Somaliland introduce biometric voter verification.’ ScienceDirect (Online service) Publisher: Elsevier Advanced Technology. Vol. 2015, no. 10, 1 Oct. 2015, pp. 3-12, doi:10.1016/S0969-4765(15)30151-*
10. *M., de Gennaro, G., de Pinto, V., Loiotile, A.D., Lovascio, S and Penza, M. (2011) ‘Odour detection methods: olfactometry and chemical sensors. Sensors (Basel)., 11(5), 5290–5322, 2011. doi: 10.3390/s110505290 Catalini, C. and Joshua, S. (2016) ‘Some Simple Economics of the Blockchain’. Rotman School of Management Working Paper No. 2874598; MIT Sloan Research Paper No. 5191-16, 2016. Available at: http://dx.doi.org/10.2139/ssrn.2874598*
11. *Cetingul, H. E., Yemez, Y. Erzin, E and Tekalp, A. M. (2006) ‘Discriminative Analysis of Lip Motion Features for Speaker Identification and Speech-Reading’, IEEE Trans. Image Processing, vol.15, no.10, pp.2879–2891. Chaum, D., Carback, R., Clark, j., Essex, A., Popoveniuc, S., Rivest, R., Ryan, P. Y. A., Shen, E., Sherman, A and Vora, P. (2009) ‘Scantegrity II: End-to-end verifiability by voters of optical scan elections through confirmation codes’, IEEE Transactions on Information Forensics and Security, 4(4):611–627.*
12. *Christidis, K. and Devetsiokiotis, M. (2016) ‘Blockchains and Smart Contracts for the IoT’, In IEEE Special Section on the Plethora of Research in Internet of Things (IoT), 2016, Vol 4, pp. 2292-2303. DOI: 10.1109/ACCESS.2016.2566339. Clack, Bakshi,*
13. *C. V and Braine, L. (2016) ‘Smart Contract Templates: Foundations, Design Landscape and Research Directions’. Available at: https://arxiv.org/pdf/1608.00771 (Accessed 17 July 2017).*