**A PROJECT REPORT ON**

# A CLOUD SECURE STORAGE MECHANISM BASED ON DATA DISPERSION AND ENCRYPTION

Submitted in partial fulfillment of the Requirements for the award of the Degree of **BACHELOR OF TECHNOLOGY**

### in

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

|  |  |
| --- | --- |
| ***by*** |  |
| **V. RAVI VARDHAN** | **1910162** |
| **K. UMESH CHANDRA** | **1910127** |
| **M. GNANESWAR** | **1910142** |

### Under the esteemed guidance of

**Sri.U.DHANUNJAYA M.Tech**



### DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING SRI KRISHNADEVARAYA UNIVERSITY

**COLLEGE OF ENGINEERING AND TECHNOLOGY ANANTAPUR – 515003 ANDHRA PRADESH**

**2023**

**SRI KRISHNADEVARAYA UNIVERSITY COLLEGE OF ENGINEERING AND TECHNOLOGY**

**ANANTAPUR – 515003**

## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



**CERTIFICATE**

Certified that this is a bonafide record of the dissertation work entitled, **“A CLOUD SECURE STORAGE MECHANISM BASED ON DATA DISPERSION AND ENCRYPTION”,** done by **V.RAVIVARDHAN(1910162), M.GNANESWAR(1910142) K.UMESH CHANDRA(1910127)** submitted to the faculty of Computer Science and Engineering, in partial fulfillment of the requirements for the Degree of **BACHELOR OF TECHNOLOGY** with specialization in **COMPUTER SCIENCE AND ENGINEERING** from Sri Krishnadevaraya University College of Engineering and Technology, Anantapur.

Signature of the Supervisor Signature of the Head of the Department(I/C)

**Sri.U.DHANUNJAYA M.Tech Mr. P.R.RAJESH KUMAR M.Tech**

### Lecturer Of Department Of CSE Lecturer Of Department Of CSE

**SKUCET,Anantapur. SKUCET,Anantapur**

**DECLARATION**

We hereby declare that the project report entitled **“A CLOUD SECURE STORAGE MECHANISM BASED ON DATA DISPERSION”** submitted to the Department of Computer Science and Engineering, Sri Krishnadevaraya University, Anantapuramu for the partial fulfilment of the academic requirement for the degree for Bachelor of Technology in Computer Science and Engineering is an authentic record of our work carried out during the final year under the esteemed guidance of **Sri.U.DHANUNJAYA, M.Tech**, Lecturer Computer Science and Engineering Department, College of Engineering and Technology, Sri Krishnadevaraya University, Ananthapuramu.

Signature of the students

1.

2.

3.

# ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of people who made it possible whose constant guidance and encouragement crowned our efforts with success. It is a pleasant aspect that I have now the opportunity to express my gratitude for all of them.

It is with immense pleasure that I would like to express my indebted gratitude **to** my **Sri.U.DHANUNJAYA, M.Tech** in **CSE Department**, who has guided me a lot and encouraged me in every step of the project work. His valuable moral support and guidance throughout the project helped me to a greater extent. I thank him for his stimulating guidance, constant encouragement and constructive criticism which have made possible to bring out this project work.

I wish to express my deep sense of gratitude to **Mr. P.R.RAJESH KUMAR, M.Tech,** Lecturer and **Head of the Department (I/C) of Computer Science and Engineering,** for giving me the opportunity of doing the project and for providing a great support in completing my project work. I feel elated to thank him for inspiring me all the way by providing good lab facilities and helping me in providing such good environment.

I wish to convey my acknowledgment to **Dr. R.RAMACHANDRA, M.Tech., Ph.D.,** Principal, **SKU College of Engineering and Technology, Anantapuramu**, for providing such a good environment and facilities.

My special thanks to the faculty of CSE Department for giving the required information in doing my project work. Not to forget, I thank all the non-teaching staff, my friends and class mates who had directly or indirectly helped and supported me in completing my project in time.

Finally I wish to convey my gratitude to my parents who fostered all the requirements and facilities that I need.

|  |  |
| --- | --- |
| **V. RAVI VARDHAN** | **1910162** |
| **K. UMESH CHANDRA** | **1910127** |
| **M. GNANESWAR** | **1910142** |

**ABSTRACT**

Cloud storage service has shown its great power and wide popularity which provides fundamental

support for rapid development of cloud computing. However, due to management negligence and

malicious attack, there still lie enormous security incidents that lead to quantities of sensitive data

leakage at cloud storages layer From the perspective of protecting cloud data confidentiality, this

paper proposed a Cloud Secure Storage Mechanism named CSSM. To avoid data breach at the

storage layer. CSSM integrated data dispersion and distributed storage to realize encrypted, chucked

and distributed storage. In addition, CSSM adopted a hierarchical management approach and

combined user password with secret sharing to prevent cryptographic materials leakage. The

experimental results indicate that proposed mechanism is not only suitable for ensuring the data

security at storage layer from leakage, but also can storage huge amount of cloud data.

TABLE OF CONTENTS Page No

[ACKNOWLEDGEMENT 4](#_TOC_250015)

[ABSTRACT 5](#_TOC_250014)

[LIST OF FIGURES 8](#_TOC_250013)

CHAPTER 1:INTRODUCTION 9

* 1. [Introduction 9](#_TOC_250012)
  2. [Literature Survey 10-12](#_TOC_250011)
  3. [Problem Formulation 12-13](#_TOC_250010)
  4. [Objective 13](#_TOC_250009)

CHAPTER 2: SYSTEM REQIREMENTS AND SPECIFICATIONS 14

* 1. [Introduction 14](#_TOC_250008)
  2. [Hardware Requirements 14](#_TOC_250007)
  3. [Software Requirements 14](#_TOC_250006)
  4. [Software Requirements Specifications 15-20](#_TOC_250005)

[CHAPTER 3: SYSTEM ANALYSIS 21](#_TOC_250004)

* 1. [Introduction 21](#_TOC_250003)
  2. [Problem Analysis 22](#_TOC_250002)
  3. Feasibility Solution 22

CHAPTER 4:SYSTEM DESIGN SPECIFICATION 23

* 1. System Architecture 23
  2. Modules 23
  3. [Detailed Design 24](#_TOC_250001)
  4. Modules Description 25

CHAPTER 5:SYSTEM IMPLEMENTATION 26

[5.1 Model Training 26-29](#_TOC_250000)

5.3 Detecting Mask in Static Images 30-32

5.3 Detecting Mask in Live Video 33-36

### CHAPTER 6: RESULTS AND DISCUSSIONS 37

* 1. Model Training Result 37
  2. Image Detection Result 38
  3. Video Detection Result 39
  4. Limitations 39
  5. Future Scope 40

### CHAPTER 7:SCREENSHOTS 41-43

**CHAPTER 8:CONCLUSION AND FUTURE WORK** 44

* 1. Conclusion 44
  2. Future Work 44

**CHAPTER 9:REFERENCES** 45

# LIST OF FIGURES

|  |  |
| --- | --- |
| 1.2.1: Object Detection | 10 |
| 1.2.2 :without-mask & with-mask images | 11 |
| 4.1 : System Architecture | 23 |
| 6.1.1 Training Loss & Accuracy | 37 |
| 6.2.1 : Image before and After for with mask | 38 |
| 6.2.2 : Image before and After for without mask | 39 |
| 7.1 : Training Face mask Detector Model | 41 |
| 7.2 : Result after Training Model | 42 |
| 7.3 : Running mask\_image | 42 |

**CHAPTER 1**

# INTRODUCTION

### Introduction

Cloud computing has shown remarkable development in recent decades. When the storage as a service, it occupies the center stage and backbone for many applications, such as pattern recognition image forensic and forgery detection. As a result, larger volumes of data will be a part of the cloud area. In the cloud industry, Amazon Web Service (AWS) has become the de facto standard. As the core component of the OpenStack that follows this standard, Swift has become one of the most popular cloud storage mechanism. However, Opens tack Swift mechanism still faces many real security threats while providing convenient services.

According to Cloud Security Alliance’s top threat case analysis report released in 2018, two thirds of the cases will cause user data leakage, mainly due to management negligence and malicious attacks. For instance, under default configuration, OpenStack Swift mechanism typically stores data in plaintext for the sake of performance. That will lead unauthorized access to user data at storage layer. In addition, Security Report released by Open stack Vulnerability Management Team VMT, the Swift mechanism may leak user data or configuration information in virtue of security vulnerabilities.

CSSM combines data dispersion with data encryption, so that large-scale cloud data and keys would be stored in chunked cipher texts. On this basis user password and secret sharing are introduced to further protect keys security. We implemented CSSM based on Open Stack Swift mechanism and made several tests.

**1.1.2 Overview**

Our Project proposes a technique CSSM where in data dispersion with data encryption, so that large-scale cloud data and keys would be stored in chunked cipher texts. On this basis user password and secret sharing are introduced to further protect keys security.

## Literature Survey

[1] A. Bhardwaj, F. Al-Turjman, M. Kumar, T. Stephan, and L. Mostarda, ''Capturing- theinvisible (CTI): Behavior-based assaults acknowledgment in IoT-situated modern control frameworks,'' IEEE Access, vol. 8, pp. 104956-104966, 2020. Modern Control Systems screen, robotize, and work complex foundation and cycles that coordinate into basic modern areas that influence our regular routines. With the presence of systems administration and mechanization, these frameworks have moved from being committed and autonomous to brought together corporate foundation. While this has worked with the checking and generally the board utilizing conventional recognition techniques, Web Application Firewalls or Intrusion Detection Systems has uncovered the organizations exposing them to Behavior- based network safety assaults. Such goes after modify the control stream and cycles andhave the noxious capacity to change the working of those frameworks by and large. This exploration centers around the use of cycle investigation to recognize assaults inside the modern control foundation frameworks and thinks about the adequacy of mark based identification techniques. The proposed work presents an example acknowledgment calculation suitably named as "Catching the-Invisible (CTI)" to search out the secret cycle in modern control gadget logs and identify Behavior-based assaults being actedprogressively.

[2] M. Kumar, A. Rani, and S. Srivastava, ''Image crime scene investigation upheld lighting assessment,'' Int. J. Picture Graph., vol. 19, no. 3, Jul. 2019, Art. no. 1950014. PC produced pictures are thought to be a critical part in every individual's life during this period of information innovation, where people successfully possess the commercials, magazines,sites, TVs and a lot of something else. At the reason when advanced pictures assumed their part, the occasion of infringement as far as deception of information, utilization of their off-base doings at last winds up and furthermore becomes simpler with the help of picture altering application programs. To be genuine, in the event that anybody fouls up anything, the proposed strategy might be utilized for a precise recognizable proof of the fraud and along these lines the impersonations inside the computerized pictures. In existing strategies, scientists have proposed most notable types of advanced visual controls upheld source, metadata, picture replicating, joining and a lot of something else. The proposed approach is enlivened by physical science based methods and requires less human association. The introduced approach works for pictures having any assortment of articles present inside the scene, for example not just restricted to human appearances and determination of same force areas of the picture. By evaluating the lighting boundaries, the proposed strategy distinguishes the controlled article and returns point of occurrence w.r.t wellspring of brightening course. The showed result produces fabrication acknowledgment pace of 92% on an image dataset including grouped kinds of controlled pictures.

[3] J. Li, Y. Zhang, X. Chen, and Y. Xiang, ''Secure property based information sharing for asset restricted clients in distributed computing,'' Compute. Secure., vol. 72, pp. 1-12, Jan. 2018. Information sharing turns into an incredibly alluring help provided by distributed computing stages because of its comfort and economy. As a potential procedure for acknowledging finegrained information sharing, quality based encryption (ABE) has drawn wide considerations. However, most of the present ABE arrangements experience the ill effects of the impediments of high calculation upward and powerless information security, which has seriously hindered asset obliged cell phones to redo the help. the question of at the same time accomplishing fine-grainedness, high effectiveness on the data proprietor's side, and standard information classification of cloud information sharing actually stays unsettled. This paper resolves this difficult issue by proposing a substitution quality based information sharing plan appropriate for asset restricted versatile clients in distributed computing. The proposed plot kills a larger part of the calculation task by adding framework public boundaries other than moving halfway encryption calculation disconnected. furthermore, a public ciphertext test stage is performed before the unscrambling stage, which dispenses with the vast majority of calculation upward due to illconceived figure texts. For data security, a Chameleon hash work is utilized to concoct an immediate code text, which can be dazed by the disconnected ciphertexts to get a definitive online ciphertexts. The proposed conspire is demonstrated secure against adaptively chosenciphertext assaults, which is well known as a standard security thought. Broad execution examination demonstrates that the proposed plot is secure and productive.

## Problem Formulation

To create an application which identifies whether a person is wearing a face mask or not and to identify people are maintaining social distancing.

## Existing System

Compared to existing Cloud storage service has shown its great power and wide popularity which provides fundamental support for rapid development of cloud computing. However, due to management negligence and malicious attack, there still lie enormous security incidents that lead to quantities of sensitive data leakage at cloud storage layer.

## Proposed System

This paper proposed a Cloud Secure Storage Mechanism named CSSM. To avoid data breach at the storage layer, CSSM integrated data dispersion and distributed storage to realize encrypted, chucked and distributed storage. In addition, CSSM adopted a hierarchical management approach and combined user password with secret sharing to prevent cryptographic materials leakage.

## Objective

The suggested mechanism's major goal is to protect cloud storage from data breaches

caused by targeted attacks (e.g. disc replicating) or management incompetence, in the event

that crackers or a malevolent controller is easily stole user information.

# CHAPTER 2

**SYSTEM REQUIREMENTS AND SPECIFICATIONS**

## Introduction

A **System Requirements Specification (SRS)** (also known as a Software Requirements

Specification) is a document or set of documentation that describes the features and behavior

of a system or software application. It includes a variety of elements (see below) that

attempts to define the intended functionality required by the customer to satisfy their different

users.

In addition to specifying how the system should behave, the specification also defines at a

high-level the main business processes that will be supported, what simplifying

assumptions have been made and what **key performance parameters** will need to be met

by the system.

## Hardware Requirements

* + - Processor : I3 7th gen
    - Ram : 8gb
    - Hard Disk : 160gb

## Software Requirements

Operating System - Windows 7/8/9

Server side Script - HTML,CSS &JS

IDE - Pycharm

Libraries Used - Numpy,IO,OS

Technology - Python 3.6+

Frame Work - Django

Databases - MySql

## Software Requirements Specifications

**Introduction to Python:**

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics.

Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it

very attractive for Rapid Application Development, as well as for use as a scripting or glue language

to connect existing components together. Python's simple, easy to learn syntax emphasizes readability

and therefore reduces the cost of program maintenance. Python supports modules and packages, which

encourages program modularity and code reuse. The Python interpreter and the extensive standard library

are available in source or binary form without charge for all major platforms, and can be freely distributed.

Since there is no compilation step, the edit-test-debug cycle is incredibly fast Debugging Python

programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter

discovers an error, it raises an exception. Python is considered an interpreted language because

Python programs are executed by an interpreter.

**HTML:**

HTML stands for **Hyper Text Markup Language**. Itis a standard markup language for web page creation. It allows the creation and structure of sections, paragraphs, and links using HTML elements (the building blocks of a web page) such as tags and attributes.

HTML has a lot of use cases, namely:

* **Web development**. Developers use HTML code to design how a browser displays web page elements, such as text, hyperlinks, and media files.
* **Internet navigation**. Users can easily navigate and insert links between related pages and websites as HTML is heavily used to embed hyperlinks.
* **Web documentation**. HTML makes it possible to organize and format documents, similarly to Microsoft Word.

It’s also worth noting that HTML is not considered a programming language as it can’t create dynamic functionality. It is now considered an official web standard. The WWW maintains and develops HTML specifications, along with providing regular updates.

**CSS:**

CSS (Cascading Style Sheets) is used to style and lay out web pages — for example, to alter the font, colour, size, and spacing of your content, split it into multiple columns, or add animations and other decorative features. This module provides a gentle beginning to your path towards CSS mastery with the basics of how it works, what the syntax looks like, and how you can start using it to add styling to HTML.

We have put together a course that includes all the essential information you need to work towards your goal.

## Prerequisites

Before starting this module, you should have:

1. Basic familiarity with using computers and using the Web passively (i.e. looking at it, consuming the content.)
2. A basic work environment set up, and an understanding of how to create and manage files, as details Dealing with files

If you are working on a computer/tablet/other device where you don't have the ability to create your own files, you could try out (most of) the code examples in an online coding program.

## Guides

This module contains the following articles, which will take you through all the basic theory of CSS,and provide opportunities for you to test out some skills.

What is CSS?

Cascading Style Sheets allows you to create great-looking web pages, but how does it work under the hood? This article explains what CSS is with a simple syntax example and also covers some key terms about the language.

Getting started with CSS

In this article, we will take a simple HTML document and apply CSS to it, learning some practical things about the language along the way.

How CSS is structured

Now that you have an idea about what CSS is and the basics of using it, it is time to look a little deeper into the structure of the language itself. We have already met many of the concepts discussed here; you can return to this one to recap if you find any later concepts confusing.

How CSS works

We have learned the basics of CSS, what it is for, and how to write simple stylesheets. In this article, we will take a look at how a browser takes CSS and HTML and turns that into a webpage.

**JavaScript:**

JavaScript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities.

JavaScript was first known as **Live Script,** but Netscape changed its name to JavaScript, possibly because of the excitement being generated by Java. JavaScript made its first appearance in Netscape 2.0 in 1995 with the name **Live Script**. The general-purpose core of the language has been embedded in Netscape, Internet Explorer, and other web browsers.

The [ECMA-262 Specification](http://www.ecma-international.org/publications/index.html) defined a standard version of the core JavaScript language.

* JavaScript is a lightweight, interpreted programming language.
* Designed for creating network-centric applications.
* Complementary to and integrated with Java.
* Complementary to and integrated with HTML.
* Open and cross-platform

## Client-Side JavaScript

Client-side JavaScript is the most common form of the language. The script should be included in or referenced by an HTML document for the code to be interpreted by the browser.

It means that a web page need not be a static HTML, but can include programs that interact with the user, control the browser, and dynamically create HTML content.

The JavaScript client-side mechanism provides many advantages over traditional CGI server-side scripts. For example, you might use JavaScript to check if the user has entered a valid e-mail address in a form field.

The JavaScript code is executed when the user submits the form, and only if all the entries are valid, they would be submitted to the Web Server.

JavaScript can be used to trap user-initiated events such as button clicks, link navigation, and other actions that the user initiates explicitly or implicitly.

## Advantages of JavaScript

The merits of using JavaScript are

**Less server interaction** − You can validate user input before sending the page off to the server. This saves server traffic, which means less load on your server.

**Immediate feedback to the visitors** − They don't have to wait for a page reload to see if they have forgotten to enter something.

**Increased interactivity** − You can create interfaces that react when the user hovers over them with a mouse or activates them via the keyboard.

**Richer interfaces** − You can use JavaScript to include such items as drag-and-drop components and sliders to give a Rich Interface to your site visitors.

## Limitations of JavaScript:

We cannot treat JavaScript as a full-fledged programming language. It lacks the following important features

Client-side JavaScript does not allow the reading or writing of files. This has been kept for security reason.

JavaScript cannot be used for networking applications because there is no such support available.

JavaScript doesn't have any multi-threading or multiprocessor capabilities.

Once again, JavaScript is a lightweight, interpreted programming language that allows you to build interactivity into otherwise static HTML pages.

One of major strengths of JavaScript is that it does not require expensive development tools. You can start with a simple text editor such as Notepad. Since it is an interpreted language inside the context of a web browser, you don't even need to buy a compiler.

To make our life simpler, various vendors have come up with very nice JavaScript editing tools. Some of them are listed here:

**Microsoft FrontPage** − Microsoft has developed a popular HTML editor called FrontPage. FrontPage also provides web developers with a number of JavaScript tools to assist in the creation of interactive websites.

**Macromedia Dreamweaver MX** − Macromedia Dreamweaver MX is a very popular HTML andJavaScript editor in the professional web development crowd. It provides several handy prebuilt JavaScript components, integrates well with databases, and conforms to new standards such as XHTML and XML.

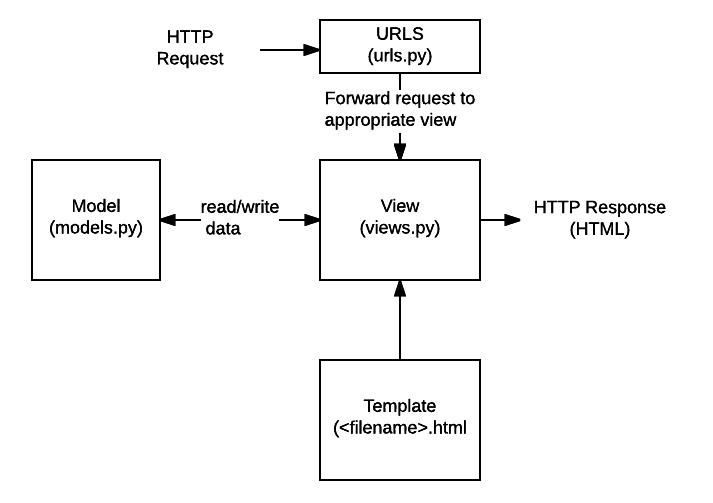
**Macromedia HomeSite 5** − HomeSite 5 is a well-liked HTML and JavaScript editor from Macromedia that can be used to manage personal websites effectively.

**Django:**

Django is a high-level Python web framework that enables rapid development of secure and maintainable websites. Built by experienced developers, Django 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 is free and open source, has a thriving and active community, great documentation, and many options for free and paid-for support.

In a traditional data-driven website, a web application waits for HTTP requests from the web browser (or other client). When a request is received the application works out what is needed based on the URL and possibly information in POST data or GET data. Depending on what is required it may then read or write information from a database or perform other tasks required to satisfy the request. The application will then return a response to the web browser, often dynamically creating an HTML page for the browser to display by inserting the retrieved data into placeholders in an HTML template.

Django web applications typically group the code that handles each of these steps into separate files:



* **URLs:** While it is possible to process requests from every single URL via a single function, it is much more maintainable to write a separate view function to handle each resource. A URL mapper is used to redirect HTTP requests to the appropriate view based on the request URL. The URL mapper can also match particular patterns of strings or digits that appear in a URL and pass these to a view function as data.
* **View:** A view is a request handler function, which receives HTTP requests and returns HTTP responses. Views access the data needed to satisfy requests via *models*, and delegate the formatting of the response to *templates*.
* **Models:** Models are Python objects that define the structure of an application's data, and provide mechanisms to manage (add, modify, delete) and query records in the database.
* **Templates:** A template is a text file defining the structure or layout of a file (such as an HTML page), with placeholders used to represent actual content. A *view* can dynamically create an HTML page using an HTML template, populating it with data from a *model*. A template can be used to define the structure of any type of file; it doesn't have to be HTML!

**MySQL:**

MySQL is popular among all databases, and is [ranked as the 2nd most popular](https://db-engines.com/en/ranking) database, only slightly trailing Oracle Database. Among open source databases, MySQL is the most popular database in use today. Known as one of the most reliable and performative databases out there, it was named after it’s founders daughter My, and is known for organizing data into one or more data tables in which data types are related to each other. These relations help structure data, as SQL is a language programmers use for creation, modification and extraction of data from a relational database.

MySQL uses standalone clients that allow users to interact with MySQL, and also to use it with other programs for applications that need relational database capabilities. MySQL's reputation for reliability has led to its inclusion in the popular LAMP stack (Linux, Apache, MySQL, Python/Perl/PHP) and is also used as the default DBMS in popular CMS options like Drupal, Joomla, phpBB, and WordPress.

**IDLE:**

IDLE is integrated development and learning environment for editing and running Python 2.x or Python 3 programs. IDLE is an integrated development environment for Python, which has been bundled with the default implementation of the language. The IDLE GUI is automatically installed with the Python interpreter. IDLE was designed specifically for use with Python. IDLE has a number of features which helps to develop Python programs including powerful syntax highlighting

# CHAPTER 3 SYSTEM ANALYSIS

## INTRODUCTION:

System Analysis is a process of collecting factual data, understand the processes involved, identifying problems and recommending feasible suggestions for improving the system functioning. This involves studying the business processes, gathering operational data, understand the information flow, finding out bottlenecks and evolving solutions for overcoming the weakness of the system so as to achieve the organizational goals. System Analysis also includes subdividing of complex process involving the entire system, identification of data store and manual processes.

The major objectives of system analysis are to find answers for each business process, what is being done, how is it being done, who is doing it, when is he doing it, why is it being done and how is it being done, who is doing it, when is he doing it, why is it being done and how can it be improved? It attempts to give birth to a new efficient system that satisfies the current needs of the user and has scope for future growth within the organizational constraints. The result of this process is logical system design. System analysis is an iterative process that continues until a preferred and acceptable solution emerges.

## PROBLEM ANALYSIS:

### Determines the present position:

The main objective of our project is to fulfil the student requirement by providing resources according to their curriculum.

### Understanding student’s requirements:

Provides required resources to the students, which helps them to increase their understanding level.

* 1. **FEASIBILITY STUDY:**

In case the system proposal is acceptable to the management, the next phase is to examine the feasibility of the system. The feasibility study is basically the test of the proposed system in the light of its workability, meeting user’s requirements, effective use of resources. These are categorized as technical, operational, economical and schedule feasibility. The main goal of feasibility study is not to solve the problem but to achieve the scope. The result is a feasibility report submitted to the management. This may be accepted or rejected. The system cycle proceeds only if the management accepts it.

# CHAPTER 4

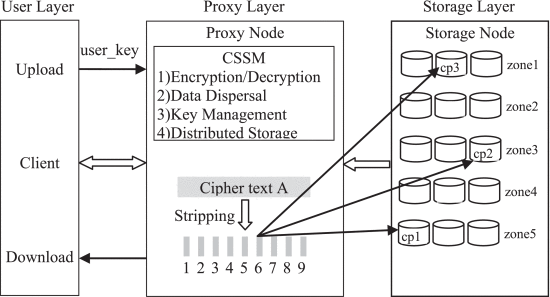
**SYSTEM DESIGN SPECIFICATIONS**

## INTRODUCTION

Based on the user requirements and the detailed analysis of the existing system the new system must be designed. This is the phase of system designing. It is the most crucial phase in the developments of a system. The logical system design arrived as a result of systems analysis is converted into physical system design. Normally, the design proceeds in two stages:

* + 1. Preliminary or General Design,
    2. Structured or Detailed Design.

## SYSTEM ARCHITECTURE



### Figure 4.1 : System Architecture

To realize primary object and properties above, this paper presents CSSM, a cloud secure storage mechanism. As shown in Figure 1, CSSM could be divided into three layers: The user layer, the proxy layer, and the storage layer. Specifically, the main functions of each layer are as follows:

To realize primary object and properties above, this paper presents CSSM, a cloud secure storage mechanism. As shown in Figure 1, CSSM could be divided into three layers: The user layer, the proxy layer, and the storage layer. Specifically, the main functions of each layer are as follows:

1. *User Layer:* This layer is deployed on the user’s machine, and the user operates (upload, download, etc.) cloud data through the client.
2. *Proxy Layer:* This layer is deployed in the cloud and composed of proxy nodes with trusted execution environments, such as Intel SGX technology [26] and ARM TrusZone technology [27]. In trusted execution environment, CSSM programs could perform as expected. CSSM in proxy layer includes four modules: data encryption/decryption, data dispersal, key management and distributed storage.
   1. *Encryption/Decryption:* This module is used to encrypt user uploaded data and decrypt user downloaded data.
   2. *Data Dispersal:* According to the data dispersal model, the cipher texts is divided into several small blocks.
   3. *Key Management:* This module is not only responsible for the generation and maintenance of the key, but also uses the hierarchical key management approach to protect the key.

## DETAILED DESIGN

### MODULES

* + - 1. Data User
      2. Cloud Server

## MODULE DESCRIPTION

**Data User:**

* Register
* After authorizer the cloud server we go the login page for user & login.
* Upload video and audio file
* View other user video and audio file and request sender
* Enter the key
* View all files
* Logout

**Cloud Server**

* Login
* View user
* View File
* View History
* View all details
* Logout

.

# CHAPTER 5 SYSTEM DESIGN IMPLEMENTATION

## Model Training

|  |
| --- |
| import os |
| import matplotlib.pyplot as plt |
| from imutils import paths |
| from tensorflow.keras.applications import MobileNetV2 |
| from tensorflow.keras.layers import AveragePooling2D |
| from tensorflow.keras.layers import Dropout |
| from tensorflow.keras.layers import Flatten |
| from tensorflow.keras.layers import Dense |
| from tensorflow.keras.layers import Input |
| from tensorflow.keras.models import Model |
| from tensorflow.keras.optimizers import Adam |
| from tensorflow.keras.applications.mobilenet\_v2 import preprocess\_input |
| from tensorflow.keras.preprocessing.image import ImageDataGenerator |
| from tensorflow.keras.preprocessing.image import img\_to\_array |
| from tensorflow.keras.preprocessing.image import load\_img |
| from tensorflow.keras.utils import to\_categorical |
| from sklearn.preprocessing import LabelBinarizer |
| from sklearn.model\_selection import train\_test\_split |
| from sklearn.metrics import classification\_report |
| dataset=r'C:\Python37\Projects\face-mask-detector\dataset' |
| imagePaths=list(paths.list\_images(dataset)) |

|  |
| --- |
| data=[] |
| labels=[] |
| for i in imagePaths: |
| label=i.split(os.path.sep)[-2] |
| labels.append(label) |
| image=load\_img(i,target\_size=(224,224)) |
| image=img\_to\_array(image) |
| image=preprocess\_input(image) |
| data.append(image) |
| data=np.array(data,dtype='float32') |
| labels=np.array(labels) |
| lb=LabelBinarizer() |
| labels=lb.fit\_transform(labels) |
| labels=to\_categorical(labels) |
| train\_X,test\_X,train\_Y,test\_Y=train\_test\_split(data,labels,test\_size=0.20,stratify=lab els,random\_state=10) |
| aug=ImageDataGenerator(rotation\_range=20,zoom\_range=0.15,width\_shift\_range=0  .2,height\_shift\_range=0.2,shear\_range=0.15,horizontal\_flip=True,vertical\_flip=True, fill\_mode='nearest') |

|  |
| --- |
| baseModel=MobileNetV2(weights='imagenet',include\_top=False,input\_tensor=Input (shape=(224,224,3))) |
| print(baseModel.summary()) |
| headModel=baseModel.output |
| headModel=AveragePooling2D(pool\_size=(7,7))(headModel) |
| headModel=Flatten(name='Flatten')(headModel) |
| headModel=Dense(128,activation='relu')(headModel) |
| headModel=Dropout(0.5)(headModel) |
| headModel=Dense(2,activation='softmax')(headModel) |
| model=Model(inputs=baseModel.input,outputs=headModel) |
| for layer in baseModel.layers: |
| layer.trainable=False |
| print(model.summary()) |
| learning\_rate=0.001 |
| Epochs=20 |
| BS=12 |
| opt=Adam(lr=learning\_rate,decay=learning\_rate/Epochs) |
| model.compile(loss='binary\_crossentropy',optimizer=opt,metrics=['accuracy']) |
| H=model.fit( |
| aug.flow(train\_X,train\_Y,batch\_size=BS), |
| steps\_per\_epoch=len(train\_X)//BS, |
| validation\_data=(test\_X,test\_Y), |
| validation\_steps=len(test\_X)//BS, |

|  |
| --- |
| epochs=Epochs |
| ) |
| model.save(r'C:\Python37\Projects\face-mask-detector\mobilenet\_v2.model') |
| predict=model.predict(test\_X,batch\_size=BS) |
| predict=np.argmax(predict,axis=1) |
| print(classification\_report(test\_Y.argmax(axis=1),predict,target\_names=lb.classes\_)) |
| # plot the training loss and accuracy |
| N = EPOCHS |
| plt.style.use("ggplot") |
| plt.figure() |
| plt.plot(np.arange(0, N), H.history["loss"], label="train\_loss") |
| plt.plot(np.arange(0, N), H.history["val\_loss"], label="val\_loss") |
| plt.plot(np.arange(0, N), H.history["acc"], label="train\_acc") |
| plt.plot(np.arange(0, N), H.history["val\_acc"], label="val\_acc") |
| plt.title("Training Loss and Accuracy") |
| plt.xlabel("Epoch #") |
| plt.ylabel("Loss/Accuracy") |
| plt.legend(loc="lower left") |
| plt.savefig(r'C:\Python37\Projects\face-mask-detector\plot\_v2.png') |

* 1. **Detecting Mask in Static Images**

From tensorflow.keras.applications.mobilenet\_v2 import preprocess\_input

|  |
| --- |
| from tensorflow.keras.preprocessing.image import img\_to\_array |
| from tensorflow.keras.models import load\_model |
| import numpy as np |
| import cv2 |
| import os |
| prototxtPath=os.path.sep.join([r'C:\Python37\Projects\face-mask- detector\face\_detector','deploy.prototxt']) |
| weightsPath=os.path.sep.join([r'C:\Python37\Projects\face-mask- detector\face\_detector','res10\_300x300\_ssd\_iter\_140000.caffemodel']) |
| net=cv2.dnn.readNet(prototxtPath,weightsPath) |
| model=load\_model(r'C:\Python37\Projects\face-mask-detector\mobilenet\_v2.model') |
| image=cv2.imread(r'C:\Python37\Projects\face-mask-detector\examples\example\_03.png') |
| (h,w)=image.shape[:2] |

|  |
| --- |
| blob=cv2.dnn.blobFromImage(image,1.0,(300,300),(104.0,177.0,123.0)) |
| net.setInput(blob) |
| detections=net.forward() |
| #loop over the detections |
| for i in range(0,detections.shape[2]): |
| confidence=detections[0,0,i,2] |
| if confidence>0.5: |
| #we need the X,Y coordinates |
| box=detections[0,0,i,3:7]\*np.array([w,h,w,h]) |
| (startX,startY,endX,endY)=box.astype('int') |
| #ensure the bounding boxes fall within the dimensions of the frame |
| (startX,startY)=(max(0,startX),max(0,startY)) |
| (endX,endY)=(min(w-1,endX), min(h-1,endY)) |
| #extract the face ROI, convert it from BGR to RGB channel, resize it to 224,224 and preprocess it |
| face=image[startY:endY, startX:endX] |
| face=cv2.cvtColor(face,cv2.COLOR\_BGR2RGB) |
| face=cv2.resize(face,(224,224)) |
| face=img\_to\_array(face) |
| face=preprocess\_input(face) |
| face=np.expand\_dims(face,axis=0) |
| (mask,withoutMask)=model.predict(face)[0] |

|  |
| --- |
| #determine the class label and color we will use to draw the bounding box and text |
| label='Mask' if mask>withoutMask else 'No Mask' |
| color=(0,255,0) if label=='Mask' else (0,0,255) |
| #include the probability in the label |
| label="{}: {:.2f}%".format(label,max(mask,withoutMask)\*100) |
| #display the label and bounding boxes |
| cv2.putText(image,label,(startX,startY-  10),cv2.FONT\_HERSHEY\_SIMPLEX,0.45,color,2) |
| cv2.rectangle(image,(startX,startY),(endX,endY),color,2) |
| cv2.imshow("OutPut",image) |
| cv2.waitKey(0) |
| cv2.destroyAllWindows() |

* 1. **Detecting Mask in Live Video**

|  |
| --- |
| from tensorflow.keras.applications.mobilenet\_v2 import preprocess\_input |
| from tensorflow.keras.preprocessing.image import img\_to\_array |
| from tensorflow.keras.models import load\_model |
| import numpy as np |
| import cv2 |
| import os |
| from imutils.video import VideoStream |
| import imutils |
| def detect\_and\_predict\_mask(frame,faceNet,maskNet): |
| #grab the dimensions of the frame and then construct a blob |
| (h,w)=frame.shape[:2] |
| blob=cv2.dnn.blobFromImage(frame,1.0,(300,300),(104.0,177.0,123.0)) |
| faceNet.setInput(blob) |
| detections=faceNet.forward() |
| #initialize our list of faces, their corresponding locations and list of predictions |
| faces=[] |
| locs=[] |
| preds=[] |
| for i in range(0,detections.shape[2]): |
| confidence=detections[0,0,i,2] |
| if confidence>0.5: |

|  |
| --- |
| #we need the X,Y coordinates |
| box=detections[0,0,i,3:7]\*np.array([w,h,w,h]) |
| (startX,startY,endX,endY)=box.astype('int') |
| #ensure the bounding boxes fall within the dimensions of the frame |
| (startX,startY)=(max(0,startX),max(0,startY)) |
| (endX,endY)=(min(w-1,endX), min(h-1,endY)) |
| #extract the face ROI, convert it from BGR to RGB channel, resize it to 224,224 and preprocess it |
| face=frame[startY:endY, startX:endX] |
| face=cv2.cvtColor(face,cv2.COLOR\_BGR2RGB) |
| face=cv2.resize(face,(224,224)) |
| face=img\_to\_array(face) |
| face=preprocess\_input(face) |
| faces.append(face) |
| locs.append((startX,startY,endX,endY)) |
| #only make a predictions if atleast one face was detected |
| if len(faces)>0: |
| faces=np.array(faces,dtype='float32') |
| preds=maskNet.predict(faces,batch\_size=12) |
| return (locs,preds) |
| prototxtPath=os.path.sep.join([r'C:\Python37\Projects\face-mask- detector\face\_detector','deploy.prototxt']) |
| weightsPath=os.path.sep.join([r'C:\Python37\Projects\face-mask-  detector\face\_detector','res10\_300x300\_ssd\_iter\_140000.caffemodel']) |

|  |
| --- |
| faceNet=cv2.dnn.readNet(prototxtPath,weightsPath) |
| maskNet=load\_model(r'C:\Python37\Projects\face-mask- detector\mobilenet\_v2.model') |
| vs=VideoStream(src=0).start() |
| while True: |
| #grab the frame from the threaded video stream and resize it |
| #to have a maximum width of 400 pixels |
| frame=vs.read() |
| frame=imutils.resize(frame,width=400) |
| #detect faces in the frame and preict if they are waring masks or not |
| (locs,preds)=detect\_and\_predict\_mask(frame,faceNet,maskNet) |
| #loop over the detected face locations and their corrosponding loactions |
| for (box,pred) in zip(locs,preds): |
| (startX,startY,endX,endY)=box |
| (mask,withoutMask)=pred |
| #determine the class label and color we will use to draw the bounding box and text |
| label='Mask' if mask>withoutMask else 'No Mask' |
| color=(0,255,0) if label=='Mask' else (0,0,255) |
| #display the label and bounding boxes |
| cv2.putText(frame,label,(startX,startY-  10),cv2.FONT\_HERSHEY\_SIMPLEX,0.45,color,2) |

**0**

|  |
| --- |
| cv2.rectangle(frame,(startX,startY),(endX,endY),color,2) |
| #show the output frame |
| cv2.imshow("Frame",frame) |
| key=cv2.waitKey(1) & 0xFF |
| if key==ord('q'): |
| break |
| cv2.destroyAllWindows() |
| vs.stop() |

# CHAPTER 6

## EXPERIMENTAL ANALYSIS AND EVALUATION

### EXPERIMMENTAL ENVIRONMENT

CSSM prototype consists of five servers, one as the proxy service node and the other four as the storage service node. Each storage server holds two hard disks, one for the system and one for the data. The specific experimental environment is shown in Table

[Table 1- 
Prototype System Experimental Environment](https://ieeexplore.ieee.org/mediastore_new/IEEE/content/media/6287639/9312710/9411835/song.t1-3075340-large.gif)

### Figure 6.1.1 Prototype System Experimental Environment

* 1. **SECURITY ANALYSIS**

CSSM enhances the security functions of proxy layer, including object encryption and dispersion, key generation and management and so on. We created a container called “encon” for storing encrypted data, tested file called “ftxt” and “ukey” as our user password. The operation steps are follows;

* ~# cat ftxt
* I write this file to do a test.
* I want to read the file from the storage device directly.
* Can I success?
* ………………………………………………..
* ~ #swift –V3 post encon –u ukey
* ~ #swift –V3 upload encon ftxt –u ukey
* ~ #swift –V3 download encon ftxt –u ukey

As shown in figure 4, file “ftxt” has been divided into 10 blocks, and each block is stored in ciphertext from. In addition, the object key box is also stored in cipher text..

[FIGURE 4. - CSSM ciphertext storage.](https://ieeexplore.ieee.org/mediastore_new/IEEE/content/media/6287639/9312710/9411835/song4-3075340-large.gif)

As for the container key box, it is processed by the secret sharing algorithm and stored in the cloud in several “coding blocks”. Therefore, in order to guarantee the security of user data, it is necessary to prove that the secret sharing algorithm can guarantee the security of these “encoded blocks”.

The secret sharing algorithm adopts the threshold value [m, n], that is, data D is encoded and converted into n blocks of data, and data D can be recovered at least through m blocks, while any data less than m blocks cannot obtain any part of the metadata information. The proof is given as follows.

* 1. **PERFORMANCE ANALYSIS AND EVALUATION**

We analyzed and evaluated the performance of CSSM mainly from three aspects: time complexities, space complexities and performance result.

**1). TIME COMPLEXITIES ANALYSIS**

CSSM uses 128-bit AES encryption, so the time cost of encryption is proportional to the size of the encrypted file. And it encrypts a file of size N in O(N). The keys are stored as object key boxes, each of which is 16 bytes in size. The size of the object key box is proportional to the number of user files in the container. Compared with the size of the user file, the time cost of encryption and decryption of the object key box is very small and can be basically ignored.

As for the index of keys, CSSM adopts keyword hierarchy derivation algorithm and user\_key set by user. In this way, a p-layer full binary tree is generated. The number of leaf nodes of the tree is n=2p−1 , and the total number of nodes is 2n-1. In the implementation, we choose n = 16, so the time cost is small and can be ignored.

**2).TIME COMPLEXITIES ANALYSIS**

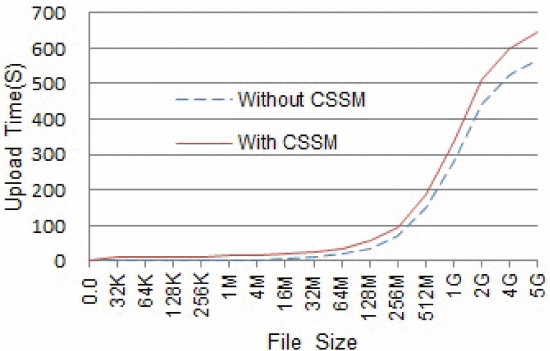
As result of adopting AES algorithm, the encrypted data is basically the same size as the original data. In terms of keys, each key length is 16 bytes. The storage space of the object key is proportional to the number of user files, and the storage space of the container key is proportional to the number of containers. Relative to the size of user files, all the storage overhead is not large.

**3).TIME COMPLEXITIES ANALYSIS**

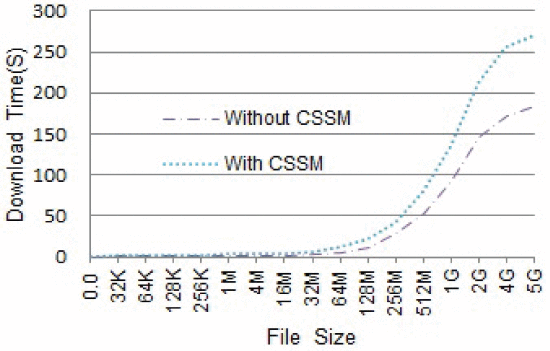
In order to evaluate the performance of CSSM, we compare the time overhead with and without using CSSM in upload and download operations based on Swift system.

In general, normal file sizes range from 32KB to 5GB. Considering the randomness of the time to complete each operation, we do 10 experiments on the same data file upload and download respectively, and the time consumed is average.

As shown in figure 5 and figure 6, additional time overhead spent on upload and download operations are illustrated respectively. By analyzing the experimental results, we can draw the following conclusions.

[[](https://ieeexplore.ieee.org/mediastore_new/IEEE/content/media/6287639/9312710/9411835/song5-3075340-large.gif)](https://ieeexplore.ieee.org/mediastore_new/IEEE/content/media/6287639/9312710/9411835/song5-3075340-large.gif)

**Figure6.1.2 : File upload time cost comparison.**

[](https://ieeexplore.ieee.org/mediastore_new/IEEE/content/media/6287639/9312710/9411835/song6-3075340-large.gif)

**Figure6.1.3 : File download time cost comparison.**

**4).The Impact of CSSM on System Overhead**

According to the experimental results, the time overhead of uploading and downloading files with CSSM increases as the file size increases. By contrast, download operations add more runtime overhead than upload operations. For example, for files of 5G size, compared with the increased running time before and after the security enhancement, the file download operation increased by 85 seconds and the file upload operation increased by 78 seconds.

The main reason is that when uploading files, the system uses multi-node concurrent operation in the data encryption and dispersion procedure. When downloading a file, the system needs to locate the location of each cipher-text fragment, and then recover from the fragments, which is a serial operation.

**5).The Functional Feature of CSSM.**

The experimental data show that the time cost of uploading and downloading files increases with the increase of file size, but the growth rate gradually slows down. For the time cost required by files of different sizes, CSSM has a better experimental effect for large files and its increase range is low. Therefore, the enhanced security features are more suitable for large files.

The experimental results show that CSSM can not only guarantee the confidential storage of data, so as to prevent the leakage of cloud data. And in terms of performance overhead, the increased time overhead is acceptable to the user.

* 1. **IMAGE DETECTION RESULT**



### Figure 6.2.1 : Image before and After for with mask

Fig.6.2.1, for the above input,the output is shown.In this case, the above subject is wearing mask.The face of the person is outlined by a green coloured border labeled as “Mask 99.88%”.The mask represents the person wearing the mask and the numbers next to mask is the confidence percentage. Fig.6.2.2: Output In the below case, the subject is not wearing a mask. So, it is highlighted as red border mentioning no mask with confidence percentage



### Figure 6.2.2 : Image before and After for without mask

* 1. **VIDEO DETECTION RESULT**

The same results obtained in image results will be obtained in Video, as the frames are processed as images in a loop.

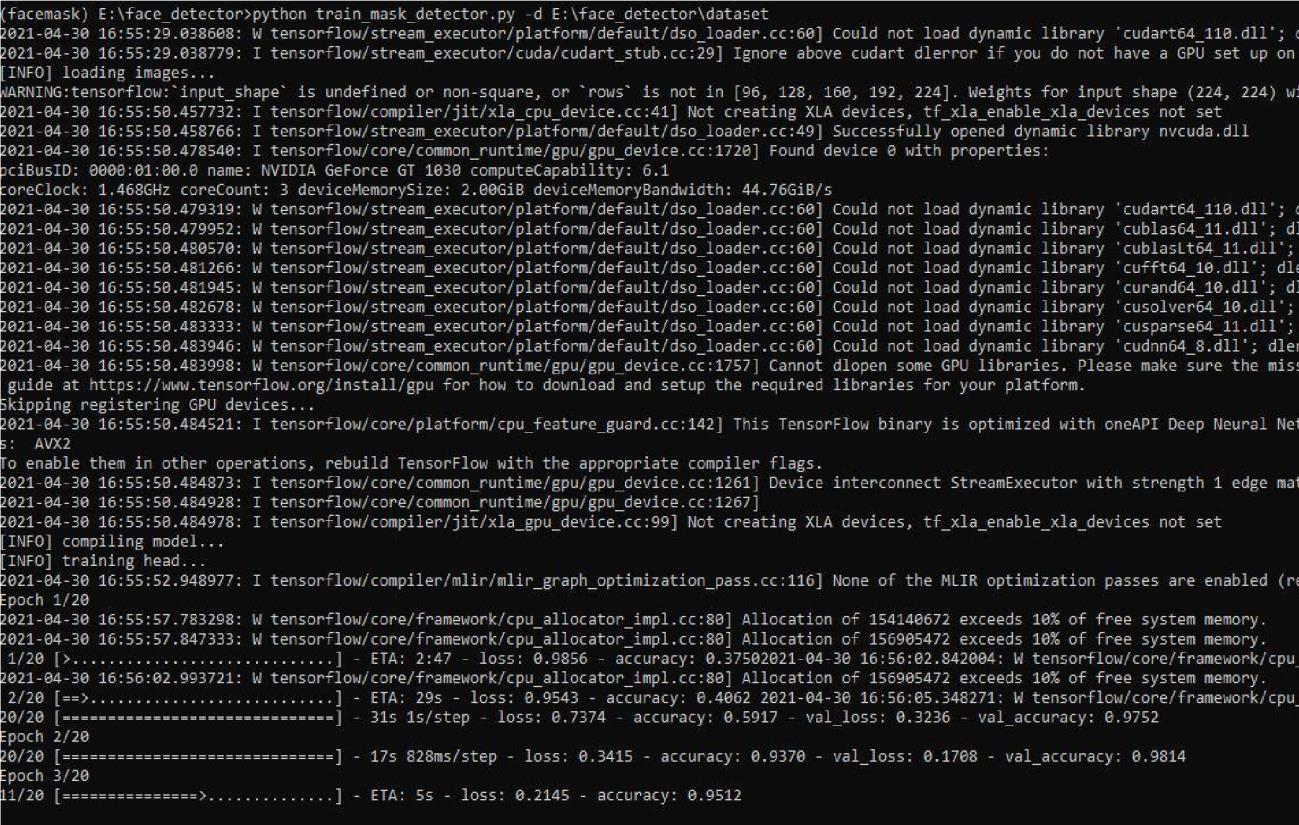
### LIMITATIONS

1. The issue with this system is that a face mask covers a piece of the face.If enough of the face is covered, the face can't be perceived, and thusly, the face mask recognition will fizzle.
2. It cannot detect people wearing masks with unique designs

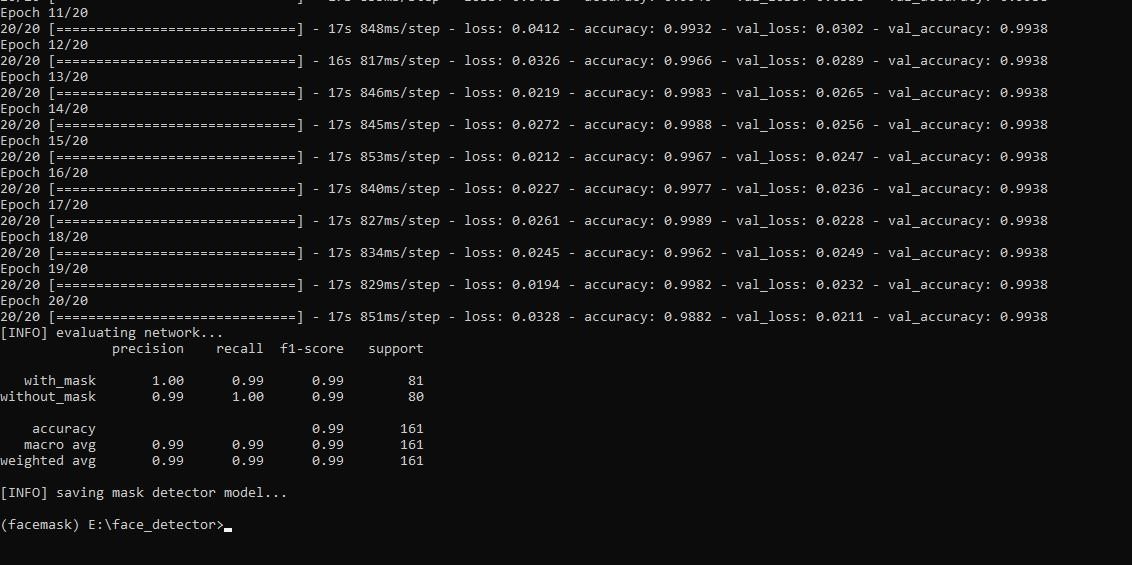
### FUTURE SCOPE

1. The first limitation can be overcome by making sure the image input is not fully obscured by mask, using advanced CV Techniques.
2. Gather dataset of people wearing different types of masks with unique designs and train the model, to improve the model accuracy.
3. This model can be used on CCTV camera monitoring, to ensure public safety.
4. This model can be used in houses along with a sound mechanism to alert people if someone is entering without wearing a mask

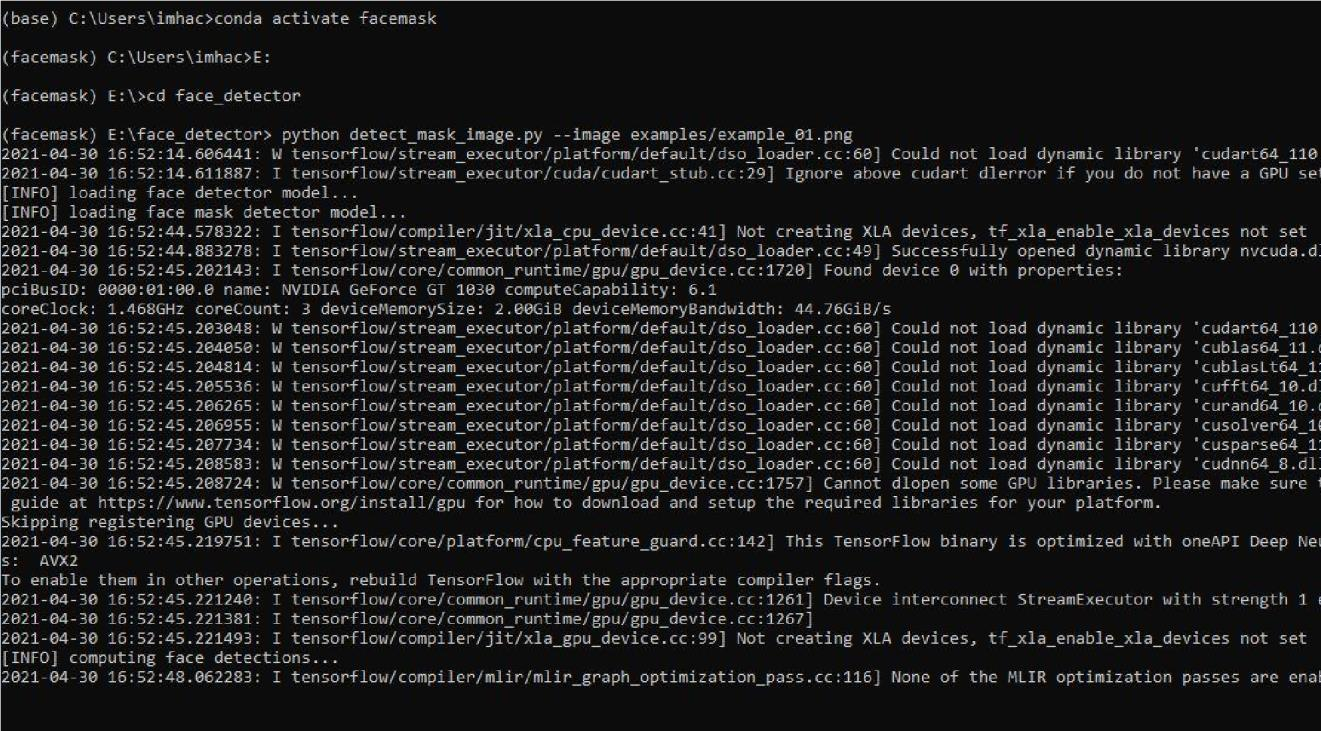
# CHAPTER 7 SCREENSHOTS



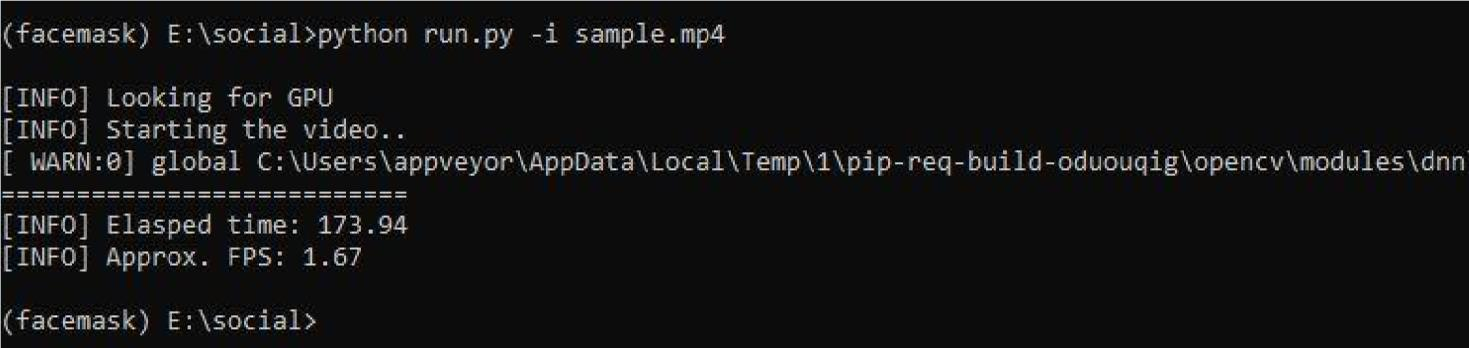
### Figure 7.1 : “Training Face mask Detector Model-1”



**Figure 7.2 : “Result after Training Model”**



**Figure 7.3 : “Running mask\_image”**



# CHAPTER 8

* 1. **Conclusion**

# CONCLUSION AND FUTURE WORK

This project describes the cloud data secure storage systems a feasible approach to solve the storage security problem especially prevention from user data leakage at cloud storage layer. CSSM could also effectively protect cryptographic materials from storage perspective with improved data security. We can achieve the better cloud storage management protocol is implemented in this system for best cloud storage management. This proposed algorithm provide the better integrity of data storage management. CSSM adopted a combined approach of data dispersal and encryption technologies, which can improve the data security and prevent attackers from stealing user data. The experimental results show that CSSM can effectively prevent user data leakage at cloud storage layer. In terms of performance, the increased time overhead of CSSM is acceptable to users. This paper provides a feasible approach to solve the storage security problem, especially prevention from user data leakage at cloud storage layer. CSSM could also effectively protect cryptographic materials from storage perspective.

## Future Works

We'll perform the extensive experiments for testing the cloud-based storage with higher volume of information and bigger size of access policies in the real cloud.

# CHAPTER10 REFERENCES

[1] A.Bhardwaj. F. Al-Turjman, M. Kumar, T. Stephan, and L. Mostarda, ''Capturing- theinvisible (CTI):

Behavior-based assaults acknowledgment in IoT-situated modern control frameworks,'' IEEE

Access, vol. 8, pp. 104956-104966, 2020.

[2] M. Kumar, A. Rani, and S. Srivastava, ''Image legal sciences upheld lighting assessment,'' Int. J. Picture Graph., vol. 19, no. 3, Jul. 2019, Art. no. 1950014.

[3] A. Bhardwaj, F. Al-Turjman, M. Kumar, T. Stephan, and L. Mostarda, ''Capturing- theinvisible (CTI):

Behavior-based assaults acknowledgment in IoT-situated modern control frameworks,'' IEEE Access,

vol. 8, pp. 104956-104966, 2020.

[4] M. Kumar, A. Rani, and S. Srivastava, ''Image legal sciences upheld lighting assessment,'' Int. J. Picture

Graph., vol. 19, no. 3, Jul. 2019, Art. no. 1950014

[5] M. Kumar, S. Srivastava, and N. Uddin, ''Image measurable upheld lighting assessment,'' Austral. J.

Measurable Sci., vol. 51, no. 3, pp. 243-250, Aug. 2017.

[6] J. Li, Y. Zhang, X. Chen, and Y. Xiang, ''Secure quality based information sharing for resource limited

clients in distributed computing,'' Comput. Secur., vol. 72, pp. 1-12, Jan. 2018.

[7] Y. Zhang, X. Chen, J. Li, D. S. Wong, H. Li, and I. You, ''Ensuring quality security assurance and quick unscrambling for reevaluated information security in portable distributed computing,'' Inf. Sci., vol. 379.