**CHROME EXTENSION FOR DETECTING**

**PHISHING WEBSITES**

A PROJECT REPORT

Submitted By

**MUBEENA NAZAR**

**(TVE20MCA-2041)**

##### to

the APJ Abdul Kalam Technological University

in partial fulfillment of the requirements for the award of the degree

##### of

Master of Computer Applications



Department of Computer Applications

College of Engineering

Trivandrum-695016

MARCH 2022

## Declaration

I undersigned hereby declare that the project report titled **”Chrome Extension for Detecting Phishing Websites”** submitted for partial fulfillment of the requirements for the award of degree of Master of Computer Applications of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by me under supervision of Smt. Minu R Nath, Assistant Professor. This submission represents my ideas in our words and where ideas or words of others have been included. I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity as directed in the ethics policy of the college and have not misrepresented or fabricated any data or idea or fact or source in our submission. I understand that any violation of the above will be a cause for disciplinary action by the Institute and/or University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title.

Place : Trivandrum **MUBEENA NAZAR**

Date : 28/02/2022

**DEPARTMENT OF COMPUTER APPLICATIONS**

COLLEGE OF ENGINEERING

TRIVANDRUM



**CERTIFICATE**

This is to certify that the report entitled **Chrome Extension for Detecting Phishing Websites** submitted by **MUBEENA NAZAR (TVE20MCA2041)** to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the Degree of Master of Computer Applications is a bonafide record of the project work carried out by her under my guidance and supervision. This report in any form has not been submitted to any University or Institute for any purpose.

Minu R Nath Prof. Deepa S S

(Project Guide) Head of the Deptartment

## Acknowledgement

If words are considered as symbols of approval and tokens of acknowledgement, then let words play the heralding role of expressing our gratitude.

First and for most I thank **GOD** almighty and to my parents for the success of this project. I owe a sincere gratitude and heart full thanks to everyone who shared their precious time and knowledge for the successful completion of my project.

I am extremely thankful to **Dr Jiji C V**, Principal, College of Engineering Trivandrum for providing me with the best facilities and atmosphere which was necessary for the successful completion of this project.

I am extremely grateful to **Prof. Deepa S S**, HOD, Dept of Computer Applications, for providing me with best facilities and atmosphere for the creative work guidance and encouragement.

I express our sincere thanks to **Asst. Prof. Minu R Nath**, Asst. Professor, Department of Computer Applications, College of Engineering Trivandrum for her valuable guidance, support and advice that aided in the successful completion of my project.

I profusely thank other Asst. Professors in the department and all other staffs of CET, for their guidance and inspirations throughout my course of study.

I owe my thanks to my friends and all others who have directly or indirectly helped me in the successful completion of this project. No words can express my humble gratitude to my beloved parents and relatives who have been guiding me in all walks of my journey.

**MUBEENA NAZAR**

**ABSTRACT**

A phishing attack is the act of sending fraudulent communications that appear to come from a valid source. Today, phishing is one of the most common forms of online fraud. The goal is to steal sensitive data like user id, password, and financial information, login information, or to install malware on the victim’s machine. ‘Phishing sites’ are a sort of safety troubles that particularly focuses on the human vulnerabilities as compared to software vulnerabilities. Phishing techniques are visual and semantically similar to those used by legitimate websites. Phishing is popular among attackers, since it is easier to trick someone into clicking a malicious link which seems legitimate than trying to break through a computers defence system.

Phishing websites can be detected using a database that contains blacklisted websites and the phishing links associated with them, which can then be compared to the entered or visited link to determine whether or not the link was in the database. However, with heuristic exhaustive search, there will be a problem if the phishing link is not in the user's database and the user will visit the website. The goal of this research project is to create a Google Chrome extension that will detect phishing websites by connecting to the databases of several online phishing detector websites that will be regularly updated. The extension also provides a report to the user stating the reason why the corresponding website was blacklisted from multiple databases in addition to giving the user an alert.

# Contents

##### Introduction v

1. Problem Definition and Motivation 3
2. Literature Review 4
3. Requirement Analysis 5
   1. [Purpose 5](#_TOC_250021)
   2. [Overall Description 6](#_TOC_250020)4
      1. [Software Requirements 6](#_TOC_250019)
      2. [Hardware Requirements 6](#_TOC_250018)
   3. [Technologies Used 6](#_TOC_250017)
      1. [Python 6](#_TOC_250016)
      2. [Tensorflow 7](#_TOC_250015)
      3. [OpenCV 7](#_TOC_250014)
      4. [LabelImg 7](#_TOC_250013)
   4. [Functional Requirements 7](#_TOC_250012)
   5. [Non Functional Requirements 7](#_TOC_250011)
4. System Design 8
5. Coding 11
6. Implementation 12
   1. [Pseudocode 12](#_TOC_250010)
      1. [Capturing Image 12](#_TOC_250009)
      2. [Training and Testing 13](#_TOC_250008)

CONTENTS

* + 1. [Detection 13](#_TOC_250007)

1. Testing 14
   1. [Testing and various types of testing used. 14](#_TOC_250006)
      1. [Unit Testing 15](#_TOC_250005)
      2. [Integration Testing 16](#_TOC_250004)
      3. [System Testing 17](#_TOC_250003)
2. Results and Discussion 18
   * 1. [Advantages 19](#_TOC_250002)
     2. [Limitations 19](#_TOC_250001)
3. Conclusion and Future Scope 20
   1. [Future Scope 20](#_TOC_250000)
   2. Data set 1. 1
   3. Data set 2. 2
   4. Image collection using Webcam. 9
   5. Labelling of Images. 9
   6. Detection of image. 10
   7. Unit test cases and results 15
   8. Integration cases and result 16
   9. System test cases and results 17

# Introduction

Gesturing is one of the earliest forms of human communication. Nowadays, Deaf and Hard of Hearing (DHH) people are the predominant users of the officially recognized sign language which consists of alphabets, numbers, and words typically used to communicate within and outside their community. Typically, a sign language consists of; (i) manual components, and

(ii) non-manual component. Specifically, the configuration, the position, and the movement of the hands form the manual components. On the other hand, the facial expression and the body movement compose the non-manual components. Such sign language is perceived as a non-verbal communication way that is mainly intended to ease the communication for the DHH persons. However, the communication between a Deaf person and a hearing individual remains an open challenge for the community. In fact, approximately 466 million people who suffer from a moderate to profound hearing loss struggle with communication daily. In other words, deaf people cannot be considered as a linguistic minority which the language can be neglected.

A sign language includes designated hand gestures for each letter of the alphabet. These gestures are used to spell people names, places, and other words without a predefined sign. Besides, it is a common occurrence for the sign formation to resemble the shape of the written letter. Although the hand gestures exhibit some similarities due to the limited number of possible hand gestures, sign language is not universal. Specifically, there are 150 sign languages around the world. They vary based on the region/country rather than the language itself. The Malayalam Sign Language (MSL) includes 62 identical alphabet signs.

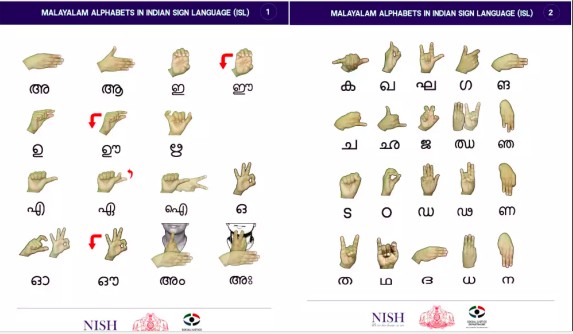
People who are unable to communicate verbally use sign language to communicate with others. One of the most important fields of research and study in computer vision is sign lan- guage. This real-time sign language recognition system was created to recognise Malayalam Sign Language gestures. So yet, no sign language recognition system for understanding gestures in Malayalam has been created. There have been various technological improvements, as well as much study, to assist the deaf and dumb. Deep learning and computer vision can also be utilised to help with the cause. This may be extremely useful for deaf and dumb people in interacting with others, as understanding sign language is not something that everyone has. Furthermore, this can be expanded to building automated editors, where a person can easily write using only their hand movements.

Figure 1.1: Data set 1.

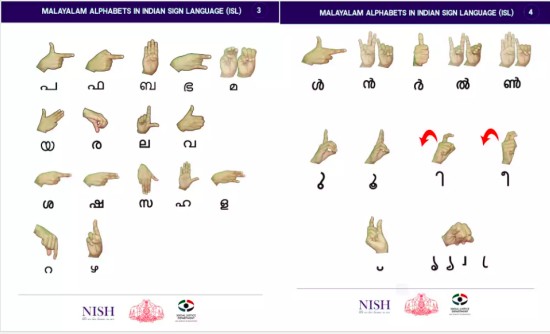


Figure 1.2: Data set 2.

# Problem Defnition and Motivation

There are a lot of folks that are unable to communicate. They are unable to communicate their messages to ordinary people. NISH (National Institute of Speech and Hearing) will assist such individuals with their education by providing sign languages. However, there is still the issue that ordinary people are unable to comprehend what they are attempting to communicate. It is the driving force behind the development of this system. For object collecting, training and testing data, and picture detection, we use OpenCV, TensorFlow, and LabelImg. The core technology behind it is deep learning.

Because deaf individuals communicate via hand signals, regular folks have a hard time un- derstanding what they’re saying. As a result, systems that identify various signs and deliver information to ordinary people are required.

# Literature Review

SLR is a prominent study topic, but despite its popularity, it is rarely used on a daily basis; this is owing to the complexity and numerous resources required. Through an examination of the methodology and models used to create a functional model of any sign-language translator from multiple sources, we examined numerous strategies that may be utilised to develop an automated sign-language translator in this literature study. The goal of this research is to look at different approaches to use Artificial Intelligence technology to enhance the currently unavailable automated Malayalam Sign Language translator. During this time, we discovered a number of study articles. The study revealed that each of the selected research studies produced reasonable findings; nevertheless, they are not flawless, since each research has its own set of strengths and shortcomings.There are various ways that may be suited for our desire to develop a usable Sign Language Translator, such as employing a regular video camera to collect data and either a Convolutional Neural Network or a Support Vector Machine to classify the input.

This study focuses on the recent advancement of technology that allow persons with speech impairments to speak readily and often with regular people. The work done thus far for development has included smart gloves, Android applications, and techniques such as Convolution Neural Networks, Gaussian filtering, HMM, voice to text, video to Text then to Speech, and so on. The system was put to the test in real time with several sample motions, and it was discovered that the average recognition rate was 99 percent. As a consequence, we decide that this effort will proceed based on the correctness of the test results.

# Chapter 2

# Requirement Analysis

**PURPOSE**

**The purpose of this project is to develop a chrome extension that detects phishing websites by connecting to an updated database. This system makes browsing more secure.**

The main functions of the proposed system are:

1. Sending an alert
2. Providing report
3. Comparison of result from multiple databases
4. Cross-checking between multiple databases

**2.1 HARDWARE REQUIREMENTS**

* **Computers equipped with a Pentium 4 processor or higher**
* **Approximately 100MB of free hard drive space**
* **128MB of RAM**
* **Web Browser**
* **Internet Connection**

**2.2 SOFTWARE REQUIREMENTS**

* **WINDOWS 7 or higher**
* **Python 3.6.0 or higher**
* **Visual Studio Code**
* **Django**
* **HTML**
* **JavaScript**

**2.3 FUNCTIONAL REQUIREMENTS**

**The chrome extension warns the user whether website is phishing or not. The extension should adhere to the following requirements:**

* **It should be able to alert the user of the website's safety.**
* **It should be faster and easier to use.**
* **It should provide a report that specifies the reasons for categorizing a website as blacklisted.**
* **It should be able to detect newly created phishing sites, and keep up with emerging phishing techniques.**

**2.4 NON-FUNCTIONAL REQUIREMENTS**

**2.4.1 User Interface**

**There must be a simple and easy to use user interface where the user should be able to quickly identify the phishing website. Input should be provided by the user, and output should be easily recognizable. The user should also receive a report regarding the results.**

**2.4.2 Performance**

**The chrome extension should be always available and should make fast detection with low false negatives. when a url of suspecting website is entered it will detect whether that website is phishing or not in quick time.**

**2.5 CONSTRAINTS AND ASSUMPTIONS**

**2.5.1 Constraints**

**Data and reports are taken from external websites.**

**Heavy techniques can’t used considering the processing power of client machines and the page load time of the website.**

**Only JavaScript can be used to develop chrome extension.**

**2.5.2 Assumption**

**The chrome extension is provided with the needed permissions in the chrome environment.**

**The user has a basic knowledge about phishing and extensions**

# Chapter 5

# System Design

Deep Learning Technology is used to detect sign language in real time. It works by en- gaging directly with the user’s language and translating it into Sign Language. It entails the following procedures:

1. Using a webcam to gather images.
2. Using LabelImg to label captured images.
3. Educate the model.

The Mobile Net architecture is used in the TensorFlow pretrained model. It’s a single convolu- tional network that learns to anticipate and categorise bounding box positions in a single run. The bounding box is an imagined rectangle that produces a collision box and acts as a point of reference for object detection. 4. The ability to detect it in real time.

The majority of the coding is done using a technique named ”Object Detection: Faster RCNN.” Only a few code tasks are required.



Figure 5.1: Image collection using Webcam.

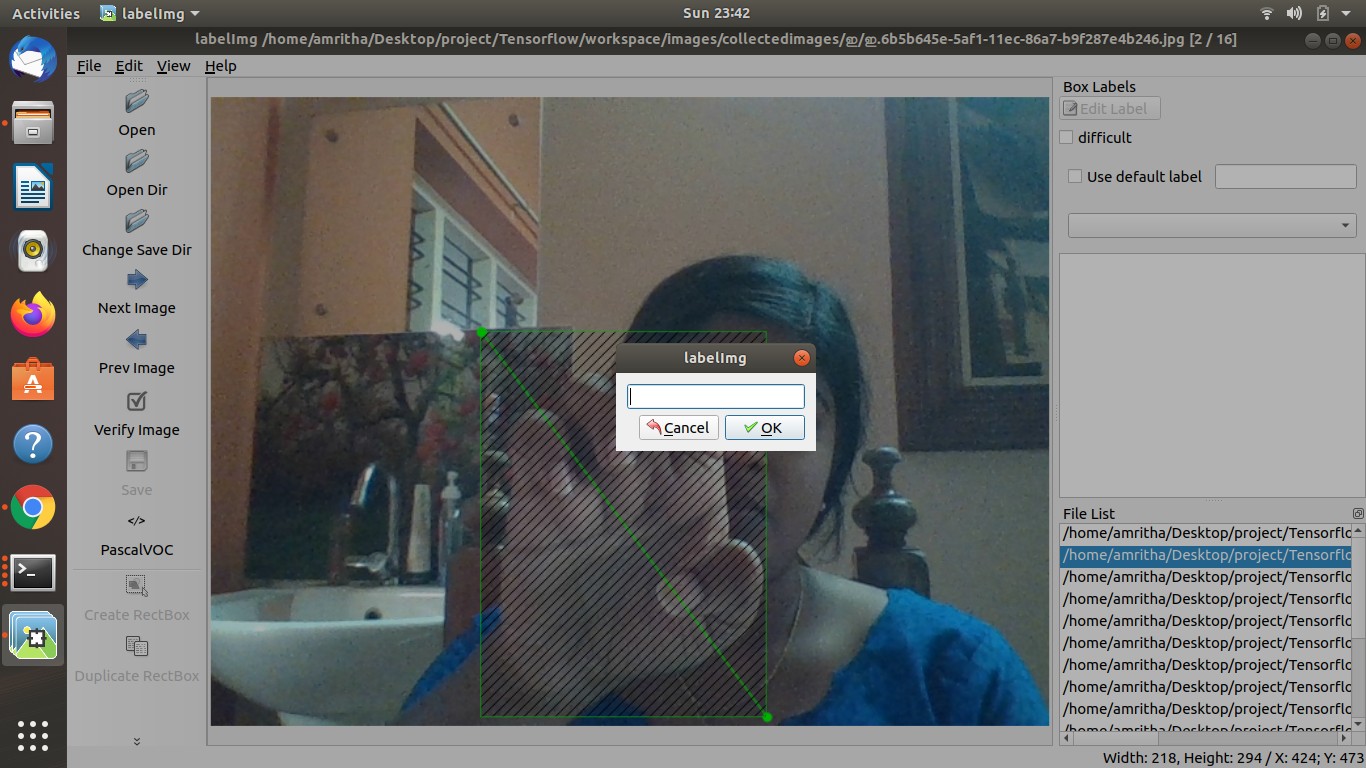


Figure 5.2: Labelling of Images.

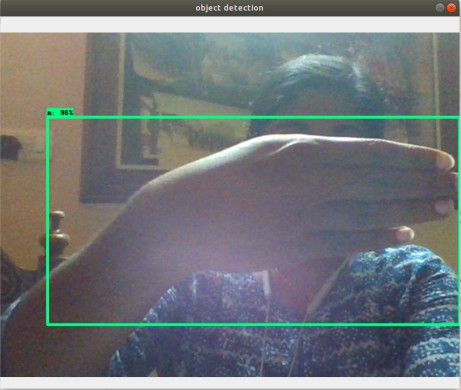


Figure 5.3: Detection of image.

# Coding

**Algorithm 1** Algorithm for Creating the model:

1: Start

2: Take an image as input.

3: Using a selected proposal approach, we obtain the Region of Interest (ROI).

4: After that, all of these areas are reshaped according to CNN’s input, and each region is handed to the ConvNet.

5: Detection.

Faster RCNN receives CNN’s feature maps and forwards them to the Region Proposal Network. RPN overlays these feature maps with a sliding window, and for each window, it constructs k Anchor boxes of various shapes and sizes: Anchor boxes are a type of fixed-size boundary box that may be found all over the picture and come in a variety of forms and sizes. RPN forecasts two things for each anchor:

* The first is the likelihood that an anchor is an object (it ignores the object’s class).
* The second is the bounding box regressor, which is used to adapt the anchors to better suit the object.

Bounding boxes of various shapes and sizes have now been passed on to the RoI pooling layer. After the RPN stage, it’s likely that some proposals will have no classes assigned to them. Each proposal can be cropped such that each proposal contains an item. The RoI pooling layer does this. For each anchor, it extracts fixed-size feature maps

# Implementation

### Pseudocode

#### Capturing Image

for label in labels:

!mkdir ’Tensorflow ’+label

cap = cv2.VideoCapture(0)

print(’Collecting images for ’.format(label)) time.sleep(5) for imgnum in range(number*imgs*) :

*ret, frame* = *cap.read*()

*imgname* = *os.path.join*(*IMAGESP ATH, label, label* +*′ .′* +*′ .jpg′.format*(*str*(*uuid.uuid*1()))) *cv*2*.imwrite*(*imgname, frame*)

*cv*2*.imshow*(*′frame′, frame*) *time.sleep*(2) *ifcv*2*.waitKey*(1)0*xFF* == *ord*(*′q′*) : *break*

*cap.release*()

*CHAPTER 7. IMPLEMENTATION*

#### Training and Testing

print(”””python /research/object*detection/modelmaintf* 2*.py − −modeldir* = */*

*−−pipelineconfigpath* = *//pipeline.config−−numtrainsteps* = 5000”””*.format*(*APIMODELP ATH, MODELP ATH, CUSTOMM ODELN AME, MODELP ATH, CUSTOMM ODELN AME*))

#### Detection

while True:

ret, frame = cap.read() image*np* = *np.array*(*frame*)

*inputtensor* = *tf.converttotensor*(*np.expanddims*(*imagenp,* 0)*, dtype* = *tf.float*32) *detections* = *detectf n*(*inputtensor*)

*numdetections* = *int*(*detections.pop*(*′numdetections′*))

*detections* = *key* : *value*[0*,* : *numdetections*]*.numpy*()*forkey, valueindetections.items*()*detections*[*′numd numdetections*

*detectionclassesshouldbeints.detections*[*′detectionclasses′*] = *detections*[*′detectionclasses′*]*.astype*(*np.in labelidoffset* = 1

*imagenpwithdetections* = *imagenp.copy*() *vizutils.visualizeboxesandlabelsonimagearray*(*imagenpwithdetections, detections*[*′detectionboxes′*]*, detec labelidoffset, detections*[*′detectionscores′*]*, categoryindex, usenormalizedcoordinates* = *True, maxboxes* 5*, minscorethresh* = *.*5*, agnosticmode* = *False*)

*cv*2*.imshow*(*′objectdetection′, cv*2*.resize*(*imagenpwithdetections,* (800*,* 600)) *ifcv*2*.waitKey*(1)0*xFF* == *ord*(*′q′*) : *cap.release*()

*break*

# Chapter 8 Testing

### Testing and various types of testing used.

System testing is the period of time during which the structure is used to ensure that it functions correctly and efficiently before live development begins. Testing is the process of putting the software into action with the goal of finding errors and missing tasks. Check to see if the objectives have been reached and the client’s needs have been addressed. Attestation of quality is a key element. The tests are completed, the results are segregated, and the standard record is kept. Everything that was thought to be incorrect turned out to be correct, and the research was completed. A test plan is created for each module using point-by-point testing approaches. The test plan depicts the unit, integration, and framework testing methods. The following are intertwined in the test scope: One of the most important goals of application framework testing is to ensure that the edge work fulfils all usable criteria, including quality requirements (Non practical prerequisites). At the end of the errand improvement cycle, the consumer should discover that the adventure fulfilled or exceeded the sum of their demands in terms of minimal essentials. Any adjustments, additions, or deletions to the requirements file, down to earth particular, or plan assurance will be documented and attempted at the highest level of important value allowed within the remainder of the project’s great importance and the experimental group’s limit.

The primary goal of testing application structures will be to detect and uncover all flaws and associated dangers, as well as to pass on each and every acknowledged issue to be addressed in a proper issue before release. The finest available strategies, approaches, work processes, and

frameworks were utilised to configure, create, execute, and manage the testing of the assignment ”Real Time Sign Language Detection” in this test approach record.

#### Unit Testing

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Sl No | Procedures | Expected result | Actual result | | | Pass or Fail |
| 1 | Data col-  lected using webcam | The sign collected using webcam | Same pected | as | ex- | Pass |
| 2 | Labeling of images | Each image is labelled and corre- sponding xml file  created. | same pected | as | ex- | Pass |
| 3 | Detection of images | Train data and de- tect it.Correct out-  put obtained. | csv file gener- ated | | | Pass |

Table 8.1: Unit test cases and results

#### Integration Testing

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Sl No | Procedures | | Expected result | Actual | result | | Pass or Fail |
| 1 | Integration | | The recorded data | Same | as | ex- | Pass |
|  | of data col- | | should be directly | pected |  |  |  |
|  | lected from | | feeded into the ten- |  |  |  |  |
|  | webcam | | sorflow module. |  |  |  |  |
|  | and Deep | |  |  |  |  |  |
|  | learning | |  |  |  |  |  |
|  | modules | |  |  |  |  |  |
| 2 | The | com- | Respective code | Same | as | ex- | Pass |
|  | mand, | | should be inserted | pected |  |  |  |
|  | context | | into the project |  |  |  |  |
|  | and values | | to implement the |  |  |  |  |
|  | should be | | desired feature. |  |  |  |  |
|  | forwarded | |  |  |  |  |  |
|  | to | the |  |  |  |  |  |
|  | module | |  |  |  |  |  |
|  | to | insert |  |  |  |  |  |
|  | code in the | |  |  |  |  |  |
|  | project. | |  |  |  |  |  |

Table 8.2: Integration cases and result

#### System Testing

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Sl No | Procedures | Expected result | Actual | result | | Pass or Fail |
| 1 | Model | The output of the | Same | as | ex- | Pass |
|  | outputs | model should be | pected |  |  |  |
|  | the cor- | based on the data |  |  |  |  |
|  | rect result | detection which |  |  |  |  |
|  | based on | is collected from |  |  |  |  |
|  | the input. | webcam. |  |  |  |  |

Table 8.3: System test cases and results

# Chapter 9 Results and Discussion

In order to conduct a comprehensive evaluation of the proposed approach, a dataset with non-uniform background and no color restrictions was collected using non-depth cameras. Specifically, our dataset includes RGB images of MSL gestures captured using web cameras from both deaf and hearing signers with different hand sizes and skin tones. One should note that there is no existing dataset. Different signers, from different nationalities, sex, and age group, performed the thirty MSL sign gestures in various backgrounds and illumination and variation according to their sign preference. This resulted in a collection of 5,000 images of size 720 x 960 x 3. The ground truth for each image consists in the label of the gesture which is the corresponding alphabet, and the coordinates of the upper left corner (x,y) and the (width, height) of the bounding box that tightly engulfs the hand gesture. Both, the labels and the bounding box coordinates are provided and used in the learning process.

It is observed that the system performs all the functionalities as expected. The system updates the Python project with valid codes based on the verbal inputs given by the user. The main aim behind this project was to ease the process of communication between common people and speech impaired people.The proposed system provides a lot of advantages over the existing system. It solves the problems of the existing system. The proposed system is more useful than the existing system in many ways

*CHAPTER 9. RESULTS AND DISCUSSION*

#### Advantages

Normal people can interact easily with speech impaired people Malayalam alphabets are detected using this.

Some words like yes, no, thanks etc. are also can be detected.

#### Limitations

It requires a lot of computational time.

Training such a huge dataset requires lot of time.

### Chapter 5

### Future Extension

We intend to integrate the IP address, email ... in the detection process in the future. Furthermore, we prefer to automate the detection process. In the future, we will build the phishing detection system as a scalable web service, integrating online learning so that new phishing attack patterns can be easily learned and improve our model accuracy with more information about the phishing attacks.

### Chapter 6

### Results and Inference

It is observed that the system performs all the functionalities as expected. The main objective of this web application is to safeguard the users from the threat of phishing. The primary objective is to safeguard the users from the threat of phishing via alerts about the security of the website and the provision of a report upon request. This way, they will be able to compare results across multiple websites and draw conclusions. Adding this feature to the context menu makes it more accessible for the users.

### Chapter 7

### Conclusion

Phishing attacks have been found to be very problematic, and a mechanism to detect them is vital in order to protect the personal information of users. Because the personal information of users can be leaked through phishing websites, it is imperative to take care of this issue. We proposed a chrome extension with more accuracy and precision when detecting phishing websites as it uses online databases in order to detect phishing websites. Meanwhile, it has a relatively low false positive rate. The extension shows a good ability to distinguish between phishing websites and non-phishing websites. By notifying users of phishing websites, our extension also provides them with the report of these sites on demand, which reduces the risk of users being scammed.

### Chapter 8

### Reference

### [1] Hongkai Chen and Mohammad Hossain, " Developing a Google Chrome Extension for Detecting Phishing Emails," 2021. [Online].Available: <https://easychair.org/publications/paper/RJLM>.

### [2] Bhavya Shah, Karan Dharamshi, Mihir Patel, Dr.Vaishali Gaikwad, " Chrome Extension for Detecting Phishing Websites," March 2020. [Online].Available: <https://www.irjet.net/archives/V7/i3/IRJET-V7I3590.pdf>.

# [3] Ms. Sophiya Shikalgar , Dr. S. D. Sawarkar , Mrs. Swati Narwane, " Detection of URL based Phishing Attacks using Machine Learning," November 2019. [Online].Available: https://www.ijert.org/detection-of-url-based-phishing-attacks-using-machine-learning.

# [4] Atharva Deshpande , Omkar Pedamkar , Nachiket Chaudhary , Dr. Swapna Borde, " Detection of Phishing Websites using Machine Learning,"May 2021. [Online].Available: <https://www.ijert.org/detection-of-phishing-websites-using-machine-learning>.

# [5] Khan A, Vuong T, Gresty D and Ahamed Khan MKA, " Phishing Detection on URLs Using Machine Learning," December 2020. [Online].Available: https://crimsonpublishers.com/nrs/fulltext/NRS.000634.php.