A

Major Project

On

MORSECODE TRANSLATOR USING EYE BLINKS

(Submitted in partial fulfillment of the requirements for the award of Degree)

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

By

C. Sainath(187R1A05C9)

G. Srikar(187R1A05E3)

I. Pranay Goud(187R1A05E8)

Under the Guidance of

MRS. G. KAVITHA REDDY

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR TECHNICAL CAMPUS

UGC AUTONOMOUS

(Accredited by NAAC, NIRF, NBA, Permanently Affiliated to JNTUH, Approved by AICTE,
New Delhi) Recognized Under Section 2(f) & 12(B) of the UGC Act.1956,
Kandlakoya (V), Medchal Road, Hyderabad-501401.

2018-22

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR
GROUP OF INSTITUTIONS
EXPLORE TO INVENT

CERTIFICATE

This is to certify that the project entitled "MORSECODE TRANSLATOR USING EYEBLINKS" being submitted by **C.SAINATH(187R1A05C9), G.SRIKAR(187R1A05E3) & I.PRANAY GOUD(187R1A05E8)** in partial fulfillment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering to the Jawaharlal Nehru Technological University Hyderabad, is a record of bonafide work carried out by him/her under our guidance and supervision during the year 2021-22.

The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

Mrs. G. Kavitha Reddy
Assistant Professor
INTERNAL GUIDE

Dr. A. Raji Reddy
DIRECTOR

Dr. K. Srujan Raju
HOD

EXTERNAL EXAMINER

Submitted for viva voice Examination held on__________________

ACKNOWLEDGEMENT

Apart from the efforts of us, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

We take this opportunity to express my profound gratitude and deep regard to my guide **Mrs. G. Kavitha Reddy,** Assistant Professor for her exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by her shall carry us a long way in the journey of life on which we are about to embark.

We also take this opportunity to express a deep sense of gratitude to Project Review Committee (PRC) **Mr. A. Uday Kiran, Mr. J. Narasimha Rao, Dr. T. S. Mastan Rao, Mrs. G. Latha, Mr. A. Kiran Kumar,** for their cordial support, valuable information and guidance, which helped us in completing this task through various stages.

We are also thankful to **Dr. K. Srujan Raju,** Head, Department of Computer Science and Engineering for providing encouragement and support for completing this project successfully.

We are obliged to **Dr. A. Raji Reddy,** Director for being cooperative throughout the course of this project. We also express our sincere gratitude to Sri. **Ch. Gopal Reddy,** Chairman for providing excellent infrastructure and a nice atmosphere throughout the course of this project.

The guidance and support received from all the members of **CMR Technical Campus** who contributed to the completion of the project. We are grateful for their constant support and help.

Finally, we would like to take this opportunity to thank our family for their constant encouragement, without which this assignment would not be completed. We sincerely acknowledge and thank all those who gave support directly and indirectly in the completion of this project.

C.SAINATH (187R1A05C9)
G.SRIKAR (187R1A05E3)
I.PRANAY GOUD (187R1A05E8)

ABSTRACT

Morse code is a method used in telecommunication/communication to encode text characters as standardized sequences of two different signal durations, called dots and dashes, or dits and dahs. Morse Code encodes the 26 Latin letters a through z, one non-Latin letter, the Arabic numerals, and a small set of punctuation and procedural signals (prosigns). There is no distinction between upper and lower case letters. Each Morse code symbol is formed by a sequence of dits and dahs. The dit duration is the basic unit of time measurement in Morse code transmission. The duration of a dah is three times the duration of a dit. Each dit or dah within an encoded character is followed by a period of signal absence, called a space, equal to the dit duration. The letters of a word are separated by a space of duration equal to three dits, and words are separated by a space equal to seven dits.

LIST OF FIGURES/TABLES

LIST OF SCREENSHOTS

1. INTRODUCTION

1. INTRODUCTION

1.1 PROJECT SCOPE

This project is titled as “MorseCode Translator” this software provides the facility to convert MorceCode into normal human readable language, this useds machine learning methodologies and EAR(Eye Aspect Ratio) to calculate dits and dahs using predefined facemesh model from mediapipe. So, this system is intended to provide an alternative form of communication for people with disabilities and to convey confidential messages.

1.2 PROJECT PURPOSE

The main purpose/objective of this project is to convert morsecode into english along with other regional Indian languages like telugu, hindi etc. This will provide the feature where user eliminates third person translator while conversing using morsecode, this model finds dits and dahs by EAR value and time constrains given by the developer, appends each character and converts English language into regional languages along with speech. With this software, a person can express themselves faster and easier. It is important to solve the problem for such people who can communicate with every person in the world.

1.3 PROJECT FEATURES

This project scheme was developed to increase the accuracy and to reduce the amount of work for people who doesn’t know morsecode. It also helps minimize the amount of time and money spent by removing third party translators. As everything is digitalized and based on data analysis, it takes less amount of time to get results. Based on the results, further action can be taken.

2. SYSTEM ANALYSIS

2. SYSTEM ANALYSIS

SYSTEM ANALYSIS

System Analysis is the important phase in the system development process. The System is studied to the minute details and analyzed. The system analyst plays an important role of an interrogator and dwells deep into the working of the present system. In analysis, a detailed study of these operations performed by the system and their relationships within and outside the system is done. A key question considered here is, “what must be done to solve the problem?” The system is viewed as a whole and the inputs to the system are identified. Once analysis is completed the analyst has a firm understanding of what is to be done.

2.1 PROBLEM DEFINITION

With this software, a person can express themselves faster and easier. It is important to solve the problem for such people who can communicate with every person in the world. As we saw in the current systems they are not easy with people who have spinal cord injuries and all. Also, the efficiencies are not up to the mark for their current systems available. Nowadays every other laptop has a camera, if not available then USB extended cameras are also available. This component of the system is most costly whereas others are inexpensive. This system will be very efficient for the people who can easily blink their eyes and have some of the other physical disability. This system can be made more accurate with pre defined face mesh model. Due to the technological advancement, its scope is infinite we can completely make it voice activated and navigated.

2.2 EXISTING SYSTEM

The manual analysis of complex-natured, is fairly a time-consuming and tedious process, and could be prone to errors, For ages the morse code has been used by many government officials during emergencies and when the normal means of communication are not available, and it is also a type of sign language used by people to communicate. We found some of the existing models like morse code translation using sound that works on the principle of sound clicks. In this method, the person will communicate with his eyes and another person need to convert it into sound (click) and that must be decoded into human-understandable language. A few existing systems have devised different representations (or) data patterns of 0's and 1's to represent a character. In this model input is taken in form of eye blinks, short blink means a dit , long blink means dah.

2.2.1 DISADVANTAGES OF EXISTING SYSTEM

- Third party translators are needed.
- Time consuming.
- Error prone or accuracy.

2.3 PROPOSED SYSTEM

In the proposed system we plan on using opencv, mediapipe to convert the morse code that is done by using different eye blinks into simple text , MediaPipe offers open source cross-platform, customizable ML solutions for live and streaming media.”, this definition is from their own website and explains what you can do with that library shortly and cleanly, they offer several other solutions that can run on different platforms and I’ll explain all of them in a different post in the future. The feature that we’ll use today is called “Face Mesh”, this solution provides us a face landmark map with the most important 468 landmarks that can be seen in a human’s face. Using that map we’ll calculate the ratio between some particular points in the face and with that information we’ll detect if the person on the camera blinked or not. To be able to detect if an eye is blinked or not we use “ear” ratio which stands for “eye aspect ratio” to be able to calculate an eye’s “ear” value we need to access 6 landmarks at one’s face. The text is then converted to other regional languages along with text and speech.

2.3.1 ADVANTAGES OF THE PROPOSED SYSTEM

- Reduces costs.
- Accurate.
- Time saving.

2.4 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. Three key considerations involved in the feasibility analysis are

- Economic Feasibility
- Technical Feasibility
- Social Feasibility

2.4.1 ECONOMIC FEASIBILITY

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require.

The following are some of the important financial questions asked during preliminary investigation:

- The costs conduct a full system investigation.
- The cost of the hardware and software.
- The benefits in the form of reduced costs or fewer costly errors.

Since the system is developed as part of project work, there is no manual cost to spend for the proposed system. Also, all the resources are already available, it gives an indication of the system is economically possible for development.

2.4.2 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

2.4.3 BEHAVIORAL FEASIBILITY

This includes the following questions:

- Is there sufficient support for the users?
- Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioral aspects are considered carefully and conclude that the project is behaviorally feasible.

2.5 HARDWARE & SOFTWARE REQUIREMENTS

2.5.1 HARDWARE REQUIREMENTS:

For developing the application, the following are the Hardware Requirements:

- Processor : intel i5 10th generation
- RAM : 4GB
- Hard Disk : 10GB
- Input Device : Standard keyword and Mouse, HD webcam
- Output Device : VGA and high resolution monitor, speaker

2.5.2 SOFTWARE REQUIREMENTS:

For developing the application, the following are the Software Requirements:

- Operating system : windows 10
- Anaconda with python3
- Spyder/Jupyter Notebook
- Open cv
- MediaPipe
- Gtts
- Play Sound
- Goggletrans

3. ARCHITECTURE

3. ARCHITECTURE

3.1 PROJECT ARCHITECTURE

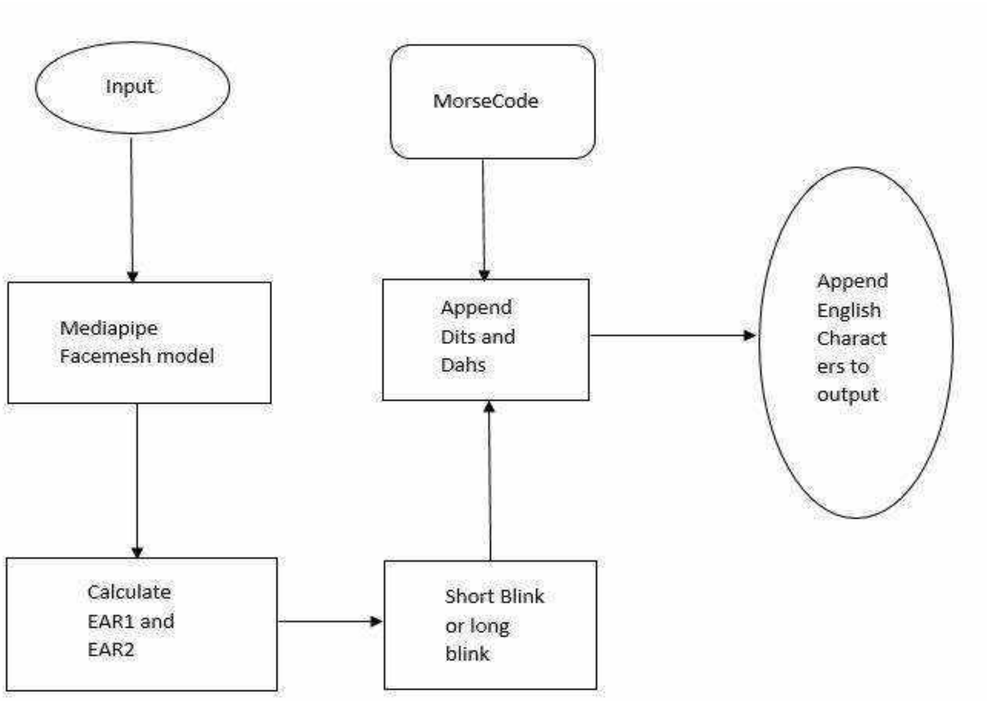


Figure 3.1: Project Architecture of Morse Code Translator

3.2 ARCHITECTURE DESCRIPTION

Input:

Input can be taken using a camera or a video, for taking input we use opencv software .

Mediapipe facemesh model:

The input is processed here, that is face mesh model is applied on face for calculating EAR

Calculate EAR1 and EAR2:

EAR1 and EAR2 are calculated for both eyes based on the formula taken.

Short blink or long blink:

If EAR value is less than threshold then it’s a blink, calculate time interval to find whether it’s a dit or dah.

Append dits and dahs:

Append all dits and dahs to form a sequence.

Morse code:

Here the actual morcecode and sequence language are present in map data structure, map the sequence taken and find appropriate alphabet.

Append English characters to output string:

Now each character is appended to the output value to form a word, this word in turn converted to other languages and speech.

3.3 MODULES DESCRIPTION

MODULE-1 (Input pre-processing)

Take input from a webcam or a video using opencv, convert the BGR input to RGB input, pass it to Media pipe Facemesh Model.

MODULE-2 (EAR calculation)

Mediapipe detects face and maps landmarks, calculate ear1 and ear 2 based on the landmarks, initialize count.

MODULE-3 (Dit or Dah)

If count is 0 then add a value else if count==10 count=0, check ear 1 and ear2, if both of them are less than threshold value then its blink, based on fps find whether its a short blink or long blink, append the value to the string.

MODULE-4 (Output)

If blinking stopped then, if there is character matching then append it to output word, array, and then convert it to the string, convert the string from English to any other language using gtts, print the output.

3.4 USECASE DIAGRAM

A use case is a set of scenarios that describing an interaction between a user and a system. A use case diagram displays the relationship among actors and use cases. The two main components of a use case diagram are use cases andactors.

Actors in use case diagram are:

- Speaker
- Listener

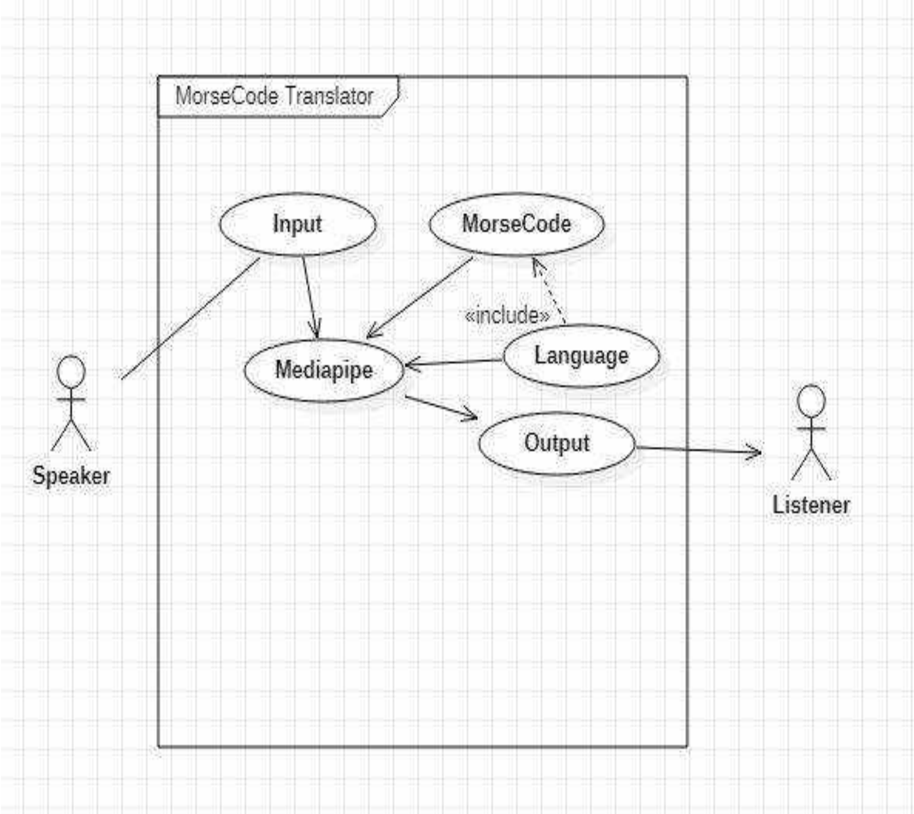


Figure 3.2: Use Case Diagram for user for MorseCode Translator

3.5 CLASS DIAGRAM

Class Diagram is a collection of classes and objects.

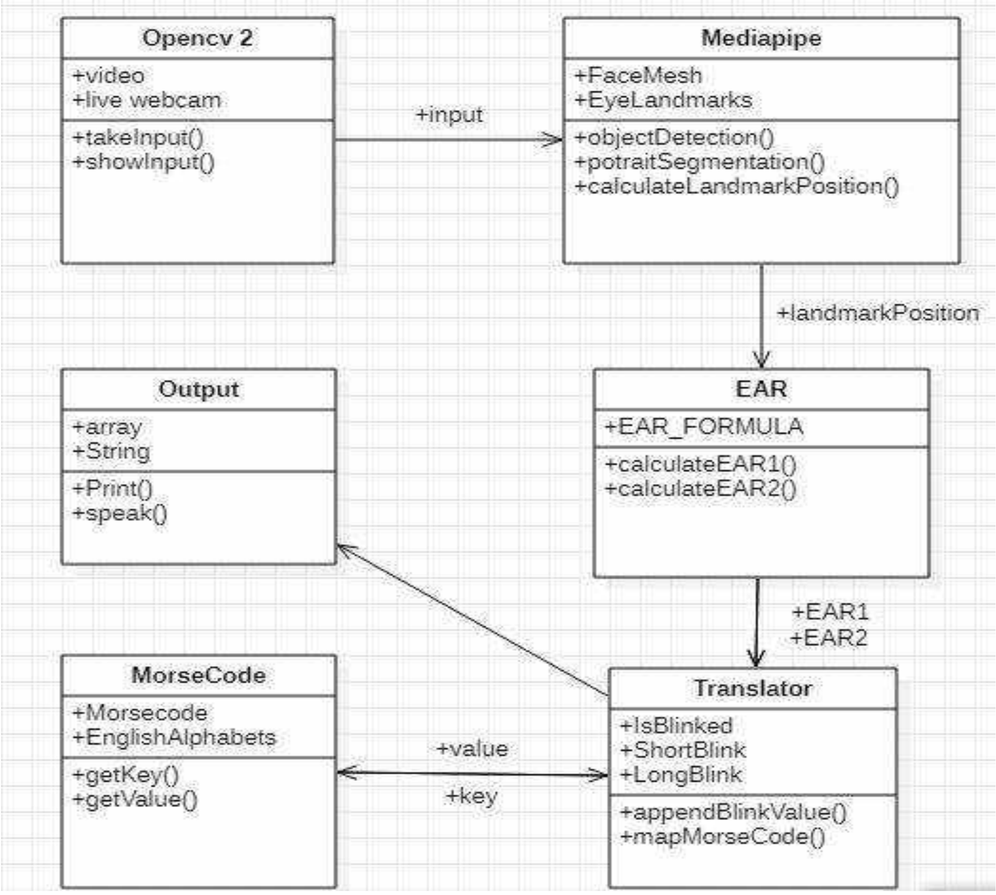


Figure 3.3: Class Diagram for MorseCode Translator

3.6 SEQUENCE DIAGRAM

Sequence diagrams in UML shows how object interact with each other and the order those interactions occur. It’s important to note that they show the interactions for a particular scenario. The processes are represented vertically and interactions are show asarrows.

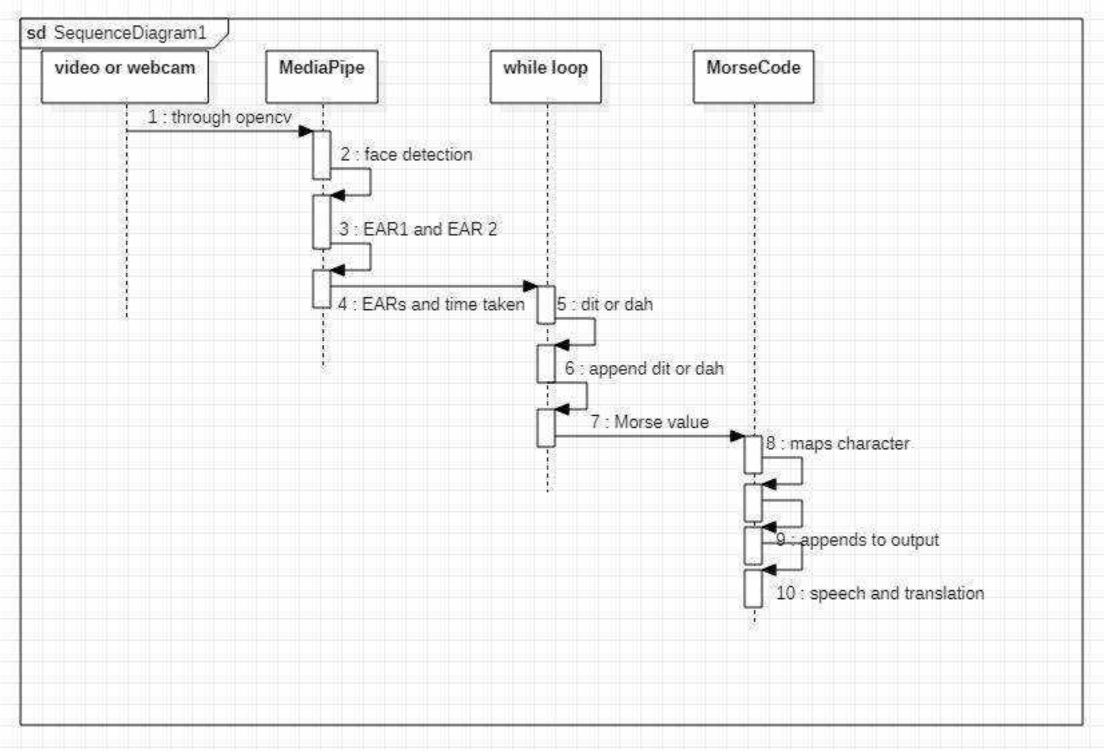


Figure 3.4: Sequence Diagram for MorseCode Translator

3.7 ACTIVITY DIAGRAM

It describes about flow of activity states.

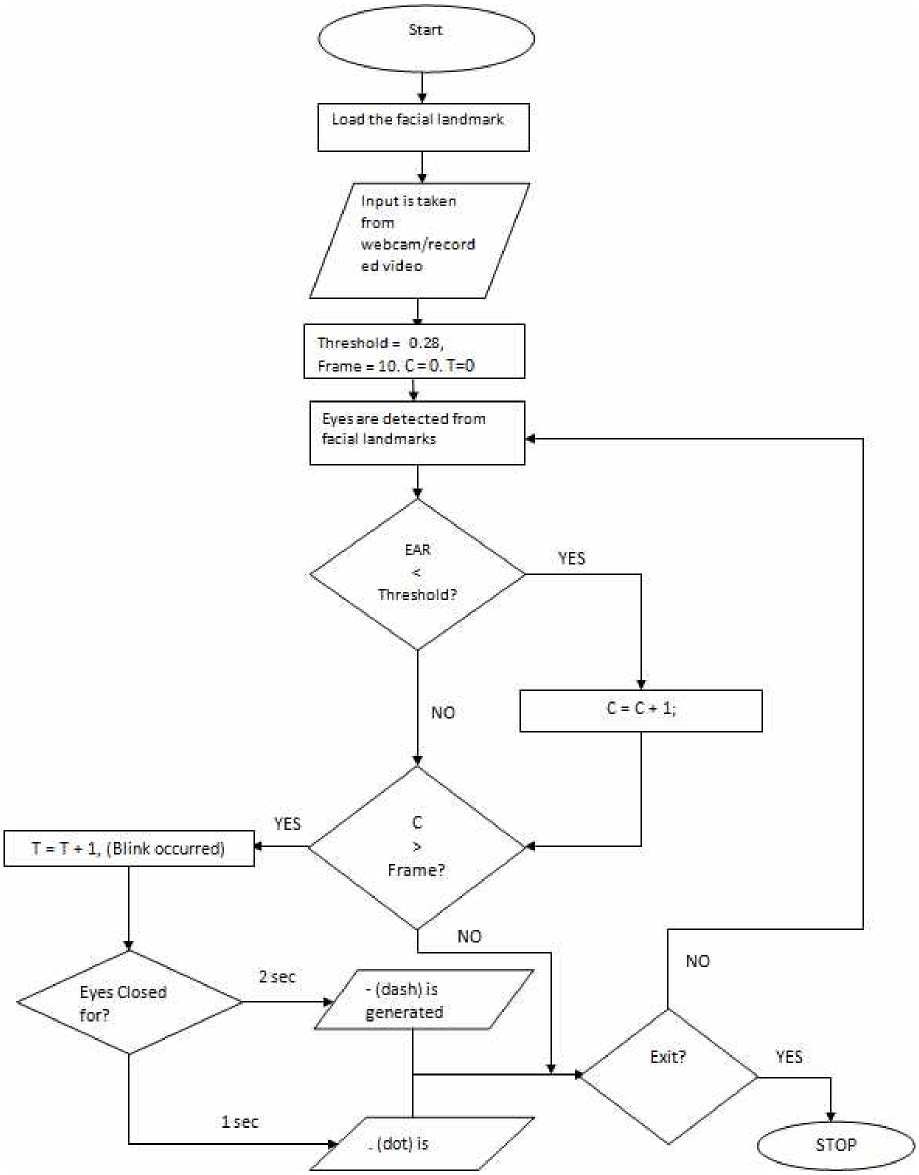


Figure 3.5: Activity Diagram for MorseCode Translator

4. IMPLEMENTATION

4.1 Code:

```
import cv2
import mediapipe as mp
import time
import utils
import math
def rescaleFrame( frame, percent=75):

    width = int(frame.shape [1]* percent/ 100)
    height = int(frame.shape [0] * percent/ 100)
    dim = (width, height)
    return cv2. resize(frame, dim)
def ld(img,results,draw=False):
    img_h,img_w=img.shape[:2]
    mesh_coord=[(int(point.x *img_w),int(point.y *img_h)) for point in
results.multi_face_landmarks[0].landmark]
    if draw :
        [cv2.circle(img,p,2,(0,255,0),-1) for p in mesh_coord]
    return mesh_coord
cap = cv2.VideoCapture(0)
pTime=0
mp_draw = mp.solutions.drawing_utils
mp_facemesh = mp.solutions.face_mesh
facemesh=mp_facemesh.FaceMesh(max_num_faces=1)
draw_spec=mp_draw.DrawingSpec(thickness=1, circle_radius=2)
ear1prev=[]
ear2prev=[]
```

```
wordArray = []

isLong = False
blinkedFor = 0

notBlinkedFor = 0

wasBlinked = False

letterArray =""

letterIs=""
MorseCode = {
"SL": "A" ,
"LSSS":"B",

"LSLS": "C",

"LSS": "D",

"S": "E",

"SSLS": "F",

"LLS": "G",

"SSSS":"H",
    "SS": "I",

    "SLLL": "J",

    "LSL": "K",

    "SLSS": "L",
    "LL": "M",

    "LS": "N",

    "LLL": "O",

    "SLLS": "P",

    "LLSL": "Q",

    "SLS": "R",

    "SSS": "S",

    "L": "T",

    "SSL": "U",
    "SSSL": "V",
    "SLL": "W",
"LSSL": "X",
"LSLL": "Y",
"LLSS": "Z"}
while True:
    success, img = cap.read()
    try:
        scaled=rescaleFrame (img, 150)
        imgRGB=cv2.cvtColor(scaled, cv2.COLOR_BGR2RGB)
        results = facemesh.process (imgRGB)
    except:
        break
    cTime = time.time()
    while cTime==0 or pTime==cTime:
        cTime=time.time()
    fps = 1/(cTime-pTime)
    pTime = cTime
    count = 0
    if results.multi_face_landmarks:
        m_c=ld(img,results,True)
    cv2.imshow("image",img)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
    if results.multi_face_landmarks:
        for faceLm in results.multi_face_landmarks:
            mp_draw.draw_landmarks(scaled, faceLm,
mp_facemesh.FACEMESH_CONTOURS, draw_spec, draw_spec)
```

```
ear1=abs((faceLm.landmark [160].x-faceLm.landmark[144].x)**2-(faceLm.landmark[160].y-faceLm.landmark [144].y)**2)+abs((faceLm.landmark[158].x-faceLm.landmark[153].x)**2-(faceLm.landmark[158].y-faceLm.landmark[153].y)**2)/abs((((faceLm.landmark[33].x-faceLm.landmark[133].x)**2)-((faceLm.landmark[33].y-faceLm.landmark[133].y)**2))

ear2= abs((faceLm.landmark [385].x-faceLm.landmark[380].x)**2-(faceLm.landmark[385].y-faceLm.landmark [380].y)**2)+abs ((faceLm.landmark[387].x-faceLm.landmark[373].x)**2-(faceLm.landmark[387].y-faceLm.landmark[373].y)**2)/abs((((faceLm.landmark[362].x-faceLm.landmark[263].x)**2)-((faceLm.landmark[362].y-faceLm.landmark[263].y)**2))

#print("ears")

#print(ear1,ear2)

if count>10:

    count=0

else:

    count=count +1

if len(ear1prev) >10:

    ear1prev[count] = ear1

    isLong = True

else:

    ear1prev.append(ear1)

if len(ear2prev) >10:

    ear2prev[count] = ear2

    isLong=True

else:

    ear2prev.append(ear2)

if isLong:

    if((ear1prev[abs(count-9)]*0.68>ear1) and (ear2prev[abs(count-9)]*0.68>ear2)):

        wasBlinked = True

        #print("blink")
```

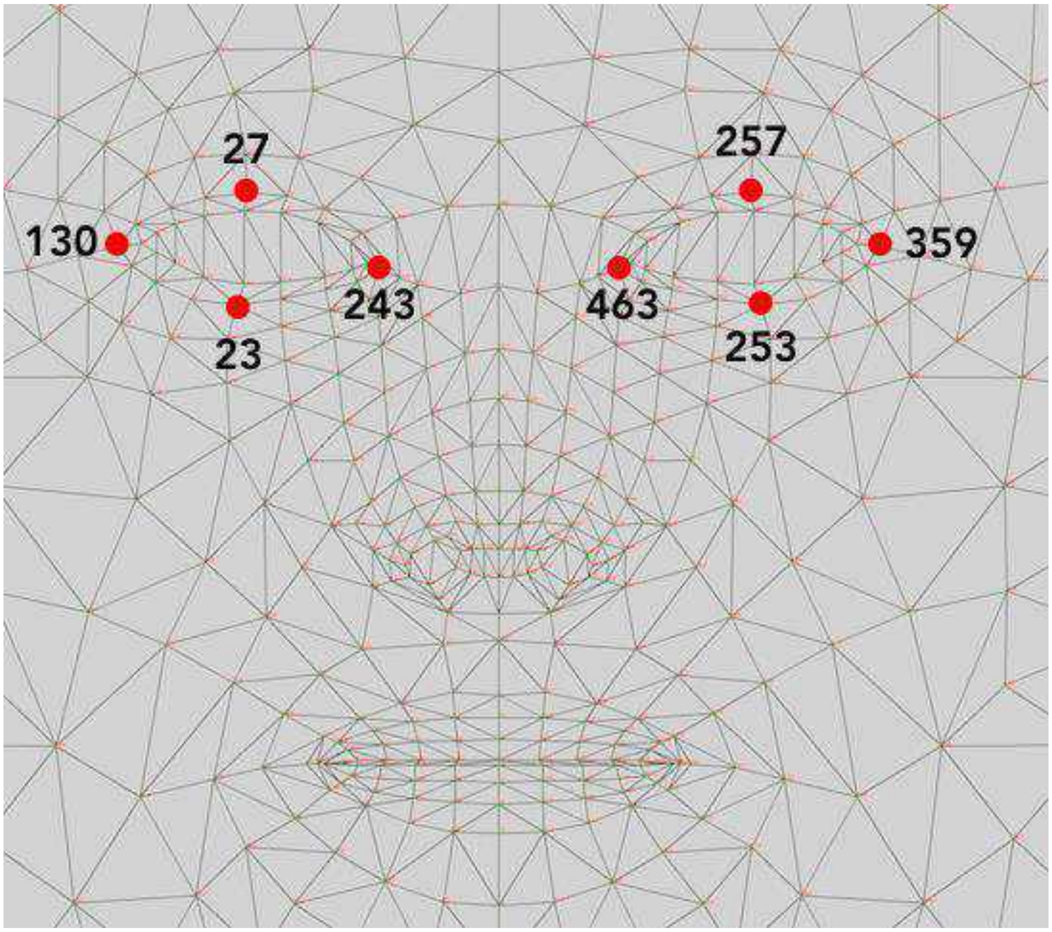
```
                    blinkedFor=blinkedFor+1
                else:
                    #print("bf",blinkedFor,fps,(fps*.7),int(fps/6.8))
                    if (blinkedFor>(fps*.7)):
                        print("LONG blink")
                        letterArray = letterArray+ "L"
                        blinkedFor=0
                    elif(blinkedFor>int(fps/6.8)):
                        print("SHORT blink")
                        letterArray=letterArray + "S"
                        blinkedFor=0
                    else:
                        notBlinkedFor=notBlinkedFor+1
                        #print("no blink")

                        if (notBlinkedFor>fps*2):
                            print(letterArray)
                            if letterArray in MorseCode:
                                letterIs=MorseCode[letterArray]
                                wordArray.append(letterIs)
                                print(wordArray)
                                letterArray=""
                            letterArray=""
                            notBlinkedFor=0
cap.release()
cv2.destroyAllWindows()
```

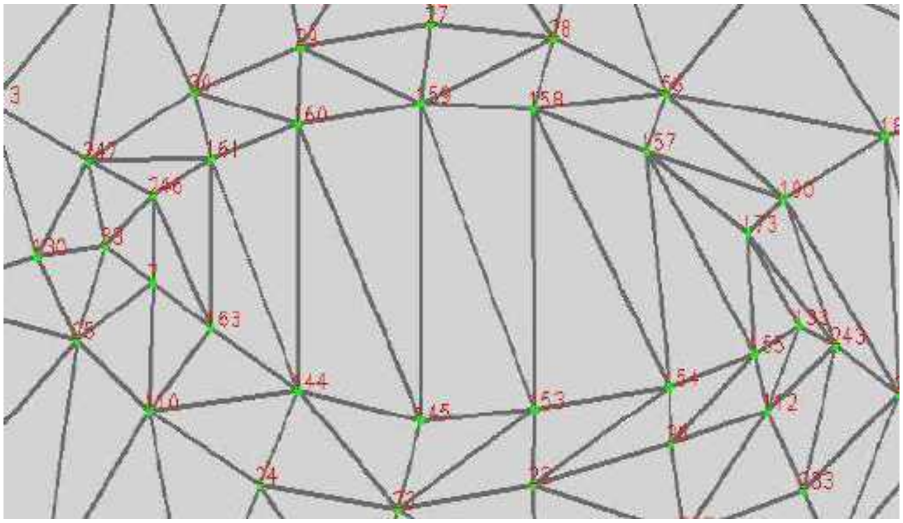
```
out=""
for i in wordArray:
    out+=i
print(out)
from googletrans import Translator
from gtts import gTTS
from playsound import playsound

translator = Translator()
t=translator.translate(text=out,src="en",dest="hi")
print(t)
speak = gTTS(text=t.text, lang="hi", slow=False)
print(speak)
speak.save("D:\\captured_voice1.mp3")
playsound("D:\\captured_voice1.mp3")
```

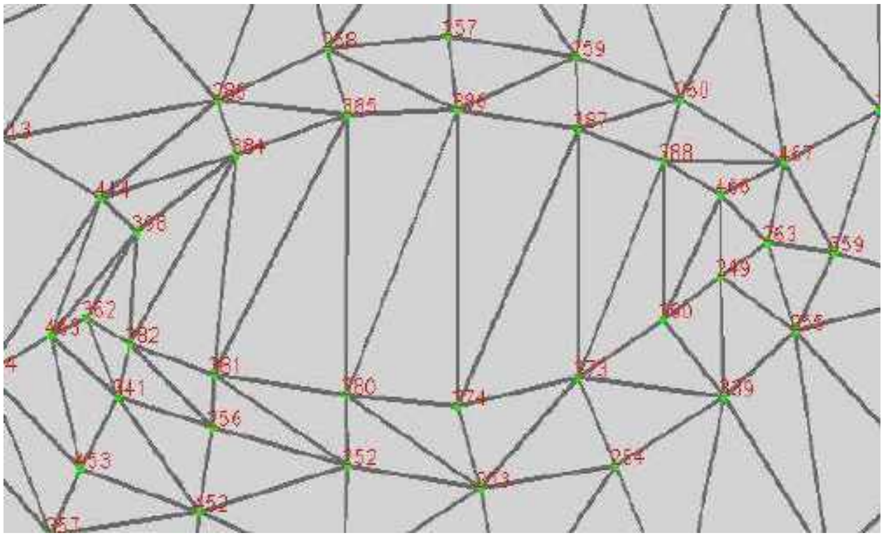
5. SCREENSHOTS



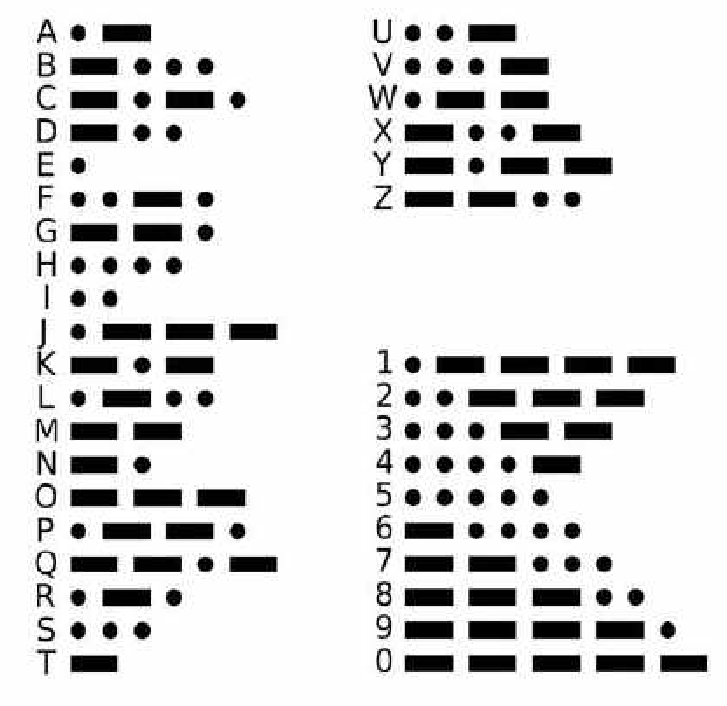
5.1 Face mesh landmarks



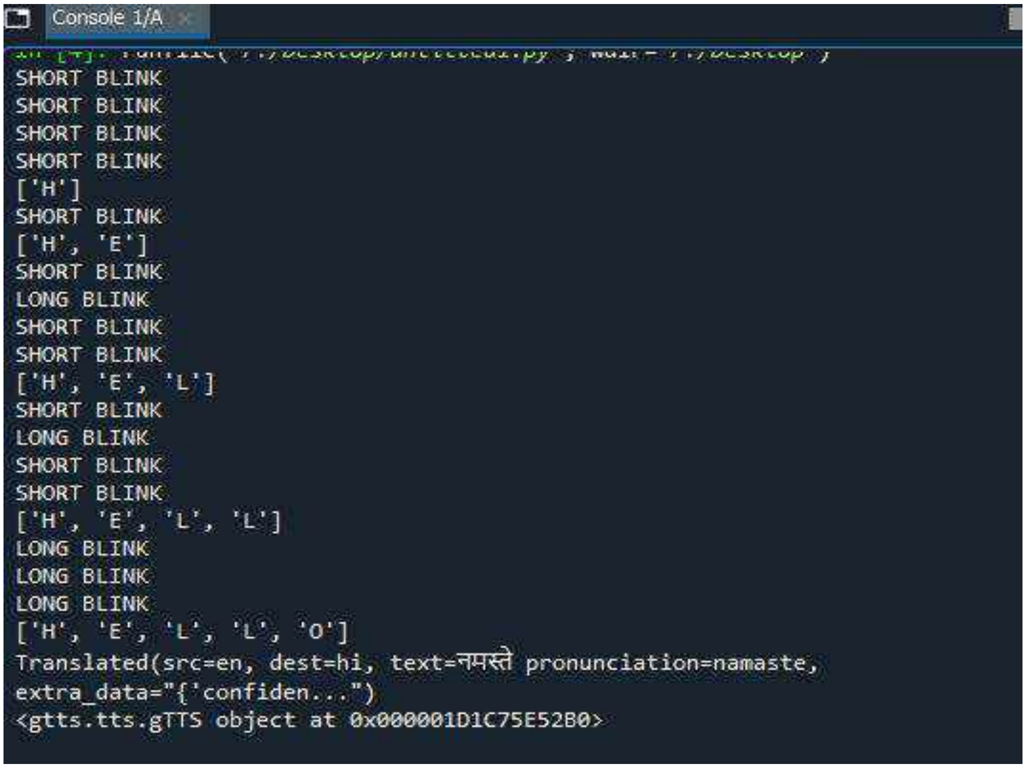
5.2 Left eye landmarks



5.3 Right eye landmarks

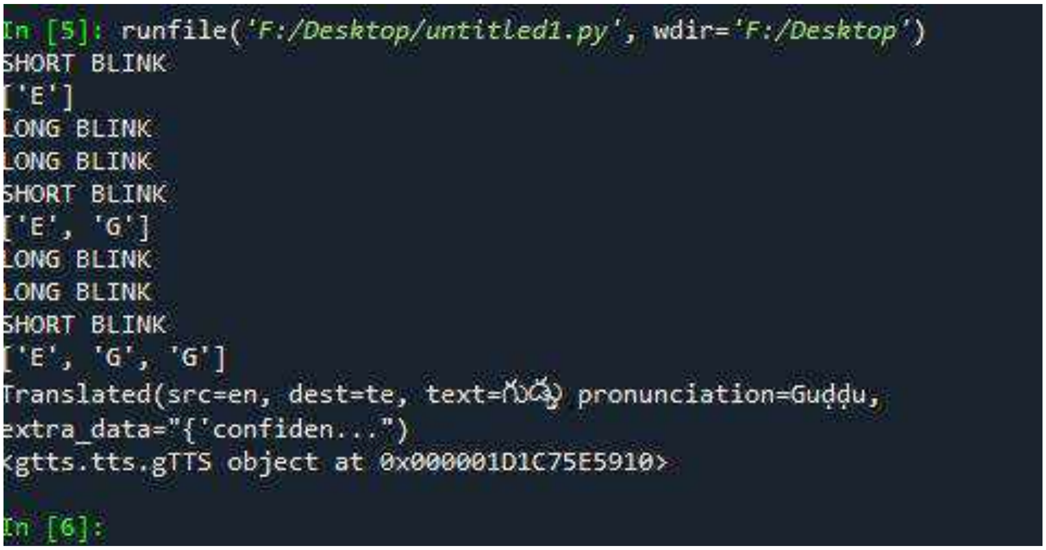


5.4 General morsecode



```
SHORT BLINK
SHORT BLINK
SHORT BLINK
SHORT BLINK
['H']
SHORT BLINK
['H', 'E']
SHORT BLINK
LONG BLINK
SHORT BLINK
SHORT BLINK
['H', 'E', 'L']
SHORT BLINK
LONG BLINK
SHORT BLINK
SHORT BLINK
['H', 'E', 'L', 'L']
LONG BLINK
LONG BLINK
LONG BLINK
['H', 'E', 'L', 'L', 'O']
Translated(src=en, dest=hi, text=नमस्ते pronunciation=namaste,
extra_data="{'confiden...")
<gtts.tts.gTTS object at 0x000001D1C75E52B0>
```

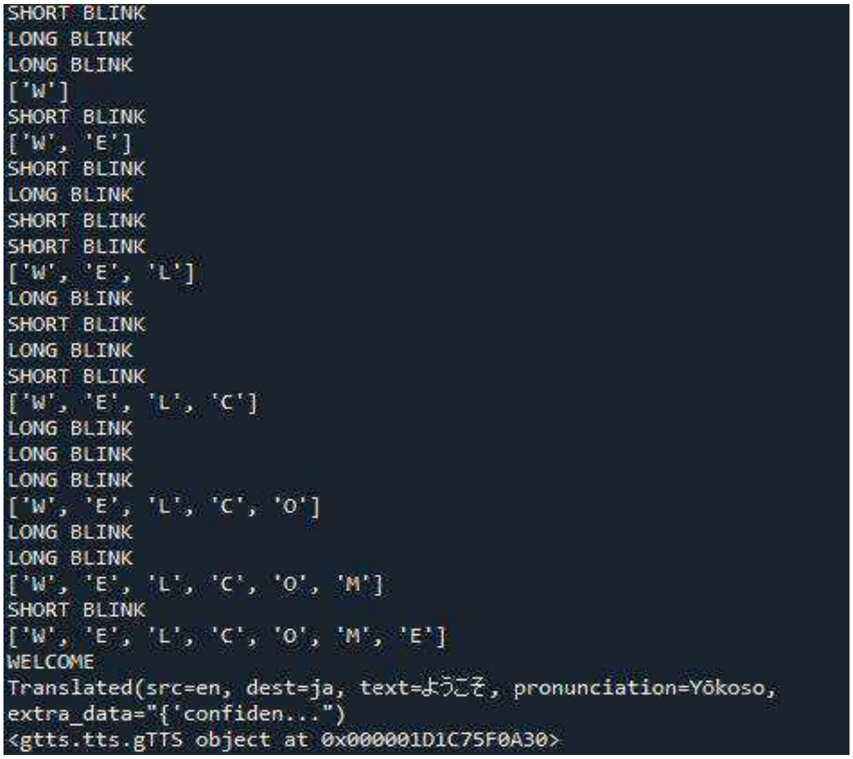
5.5 morsecode for hello, hello converted to hindi



```
In [5]: runfile('F:/Desktop/untitled1.py', wdir='F:/Desktop')
SHORT BLINK
['E']
LONG BLINK
LONG BLINK
SHORT BLINK
['E', 'G']
LONG BLINK
LONG BLINK
SHORT BLINK
['E', 'G', 'G']
Translated(src=en, dest=te, text=గుడ్డు pronunciation=Guḍḍu,
extra_data="{'confiden...")
<gtts.tts.gTTS object at 0x000001D1C75E5910>

In [6]:
```

5.6 morsecode for egg, egg converted to telugu



```
SHORT BLINK
LONG BLINK
LONG BLINK
['W']
SHORT BLINK
['W', 'E']
SHORT BLINK
LONG BLINK
SHORT BLINK
SHORT BLINK
['W', 'E', 'L']
LONG BLINK
SHORT BLINK
LONG BLINK
SHORT BLINK
['W', 'E', 'L', 'C']
LONG BLINK
LONG BLINK
LONG BLINK
['W', 'E', 'L', 'C', 'O']
LONG BLINK
LONG BLINK
['W', 'E', 'L', 'C', 'O', 'M']
SHORT BLINK
['W', 'E', 'L', 'C', 'O', 'M', 'E']
WELCOME
Translated(src=en, dest=ja, text=ようこそ, pronunciation=Yōkoso,
extra_data="{'confiden...")
<gtts.tts.gTTS object at 0x000001D1C75F0A30>
```

5.7 morsecode for welcome, welcome converted to japanese

6. TESTING

6. TESTING

6.1 INTRODUCTION TO 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, subassemblies, 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.

6.2 TYPES OF TESTING

6.2.1 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 structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at the 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.

6.2.2 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 satisfied, as shown by successful 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.

6.2.3. FUNCTIONAL TESTING

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.

The organization and preparation of functional teats are focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining identifying Business process flows; data fields, predefined processes.

6.3. TEST CASE

Test case ID	Test case name	Purpose	Input	Output
1	Test Case 1	To translate Morse code into Speak able language	. _ _ . _ _ .	EGG In Telugu (text=గుడ్డు, pronunciation=Guḍḍu), MP3 FILE
2	Test Case 2	To translate Morse code into Speak able language	. _ _ . . _ . . _ . _ . _ _ _ _ _ .	WELCOME, In Japanese (text=ようこそ, pronunciation=Yōkoso, MP3 FILE
3	Test Case 3	To translate Morse code into Speak able language	. . . . . . _ . . . _ . . _ _ _	HELLO, In Hindi (text=नमस्ते, pronunciation=namaste), MP3 FILE

6.3.1 TEST CASE DESCRIPTION

TEST CASE 1 : When we give input the code try’s to translate it into ENGLISH and then to other specified Language along with speech i.e to EGG and గుడ్డు .

TEST CASE 2 : When we give input the code try’s to translate it into ENGLISH and then to other specified Language along with speech i.e to WELCOME and ようこそ.

TEST CASE 3 : When we give input the code try’s to translate it into ENGLISH and then to other specified Language along with speech i.e to HELLO and नमस्ते.

7. CONCLUSION

7. CONCLUSION & FUTURE SCOPE

7.1. PROJECT CONCLUSION

- The existing system was quite complicated and people working on this system must have to remember a lot.
- Whereas our system has resolved the complicated issues and added predictive power to it.
- Here we translate not only with one language; we can translate to more regional languages and converting the generated text into the speech.

7.2. FUTURE SCOPE

- Can improve accuracy by improving the model.
- Can make it more dynamic by using any angle of face to the camera.
- Can be used to implement a GUI application.

8. BIBLIOGRAPHY

8. BIBLIOGRAPHY

8.1 GITHUB LINK

https://github.com/sainathchp/major-project

https://github.com/srikar5710/major-project

https://github.com/pranay-root/my-project

8.2REFERENCES

[1] https://www.jetir.org/papers/JETIR2105194.pdf

[2] https://google.github.io/mediapipe/solutions/face_mesh#python-solution-api

[3] www.wikipedia.com/

9. JOURNAL

Volume: 10 Issue: VI Month of publication: June 2022

DOI:

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue VI June 2022- Available at www.ijraset.com

Morse code Translator Using Eye Blinks

Kavitha Reddy Guda[1], Sainath Cheparthy[2], Srikar Gangipally[3], Pranay Goud Iruvuri[4]
[1, 2, 3, 4]*Department of Computer Science and Technology, CMR Technical Campus, Medchal , 501401*

Abstract: A Morse code Translator which translates Morse code into speech and text of any language chosen by the user, Morse code is given as input ,it can be either a recorded video or live feed of a person who is blinking eyes in a sequence .OpenCV is used to take Morse code input ,Mediapipe a google api detects face and maps facial landmarks, These landmarks are used to map various eye co-ordinates and then considered to formulate Eye Aspect ratios ,which in turn determines eye blink .Morse code which has been already loaded in the form of dictionary is going to map with the morse code given through video or cam ,and find the Alphabet assigned to it.

I. INTRODUCTION

Communication is a process which is essential in day-to-day life for a person, every person has needs to communicate with people, the mode of communication can be any, it can be either speech, text, signs Etcetera. People communicate using any of the modes mentioned, but some persons are disable, disability can be of many forms like deaf, dumb, blind Etcetera. Disability can be caused due to many reasons, from birth or other medical conditions so every communication mode is not Feasible for every person or the person with whom he/she is trying to communicate.

Also, some people want to communicate in secret so that they can pass messages to specified person in presence of third person who will have no knowledge about that mode, this can be done using Morse code. Morse code is a communication process that uses two distinct signal durations to encode text characters.

In the Morse code there are 26 letters from a to z, one non-Latin letter, numbers, and a small set of punctuations and procedures (signs). Morse code is not case sensitive.

Morse code is accompanied with sequence of dits and dahs. Morse code transmission is measured by dit durations. dah is generally three times longer than a dit. Normally, after each dit or dah in an encoded character, there is a signal absence, called a space, which is equal to the dit duration.

There is a space of duration equal to three dots between each letter of a word, and seven dots separate words Morse code can be memorised and sent in a form that is apparent to human senses, such as sound waves or visible light, eye blinks, so that it can be directly deciphered by people who have learned the technique. Morse code is frequently communicated through an information-carrying medium such as visible light, radio waves, or sound waves, which is keyed on and off in sequence. The current or wave is present during the dit or dah's duration and missing between dits and dah's duration. As mentioned, we are using Eye blinks to communicate through Morse code, A short blink is a dit and a long blink is a dah and sequence of the is an alphabet these alphabets together form a word. Jeremiah Andrew Denton Jr. (July 15, 1924 – March 28, 2014) was an American politician and military officer, while answering questions Denton blinked his eyes in Morse code, spelling the word "T O R T U R E" and sent a secret message through his eyes to his US NAVAL peers. This example shows how one can use Morse code in many situations. but one to receive it and understand message given in morse code must know morse code, but now one can simply use this software upload the video and convert the morse code to text, speech and also into any other language they know this software allows users to translate morse code in an efficient, user friendly and cost-effective manner.

II. LITERATURE REVIEW

Technology has a wide area which updates day by day, there are various technologies used to convert various forms of morse codes, sign languages Etcetera. there are a lot of projects which detect faces and eye blinks. Morse code translation can be done can be done using may technologies some of the projects are

1) Using CNN (Convolutional Neural Network) or MTCNN (Multi-Task Cascaded Convolutional Neural Network) which in turn uses three different layers.
2) Using openCV and tree-based algorithms.

III.PROPOSED MODEL

A. Approach

Our approach is by using mediapipe and openCV. Mediapipe is a google api offers cross-platform, customizable ML solutions for live and streaming media. There are various ML solutions available such as Face Mesh model, Object-detection, pose, hands, holistic, Box tracking and many more. Face Mesh Model is a model which is used frequently by developers. Face Mesh model allows users to map 468 3D landmarks upon a face. Bringing real-time augmented reality (AR) applications, a transform module bridges the gap between face landmark estimation and useful AR applications that are commonly used in everyday life.

Steps

1) Connect your cam to OpenCV and load the live input or give already recorded video
2) Apply mediapipe's face mesh model on input and initialize Morse code
3) Calculate initial EAR's (Eye Aspect ratios)
4) Check if the EAR's values are less than the threshold value
5) If yes that's a blink, then find out whether it's a long or short blink based on duration
6) Append dits and dahs to create a sequence
7) Map matching character for the sequence
8) Append the character to a string to form a word
9) Translate word string and convert it into speech

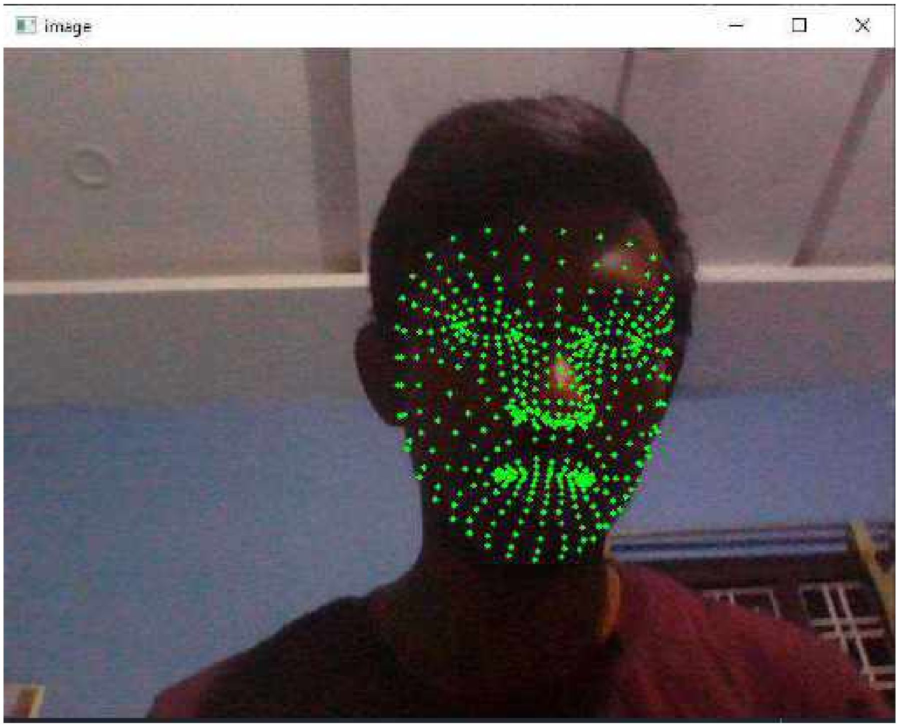


Fig 1: Live feed of a person communnicating in morse code

B. Modules

1) *EAR Formulation:* EAR (Eye Aspect Ratio) can be calculated by below formula where each point represents landmark of eye's X and Y co-ordinate. The total points for each eye is six since we are considering two vertical lines and two horizontal lines.

$$EAR=\frac{|(p1_x - p2_x)^2-(p1_y - p2y)^2|+|(p3_x - p4_x)^2-(p3_y - p4_y)^{2}}{|((p5_x - p6_x)^{2})-((p5_y - p6y)^2)|}$$

Fig 2: EAR Formula

Where , (p1,p2),(p3,p4) are 2 vertical lines anf (p5,p6) is horzontal line.

2) *Determining Blinks and No Blinks:* A comparison is made between the ear values of two eyes. the threshold value is calculated Using a formula which contains previous and current time values. if the EAR is less than Threshold then is a blink, else it isn't a blink. It is determined by the duration of the blink whether it is a long blink or a short blink.
3) *Mapping Morse Code:* The dits and dahs are then attached once the blinks have been found, this sequence maps with the Morse code in the software if it's a match then mapped character is append to word string and when user stops the program the word string is converted to speech and text as selected by the user.

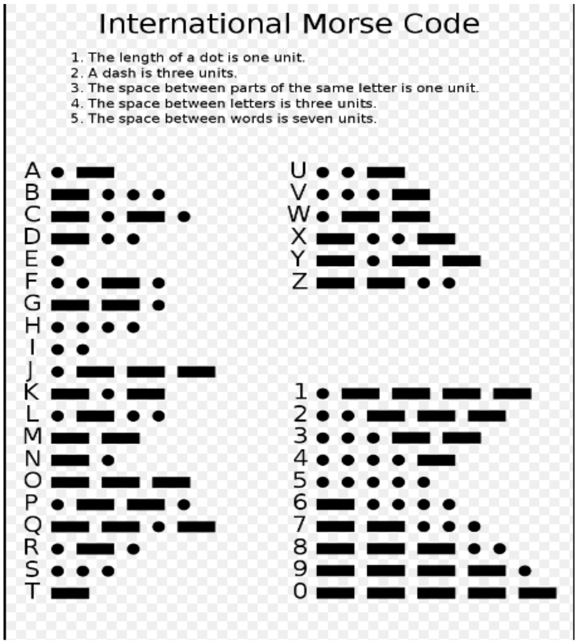
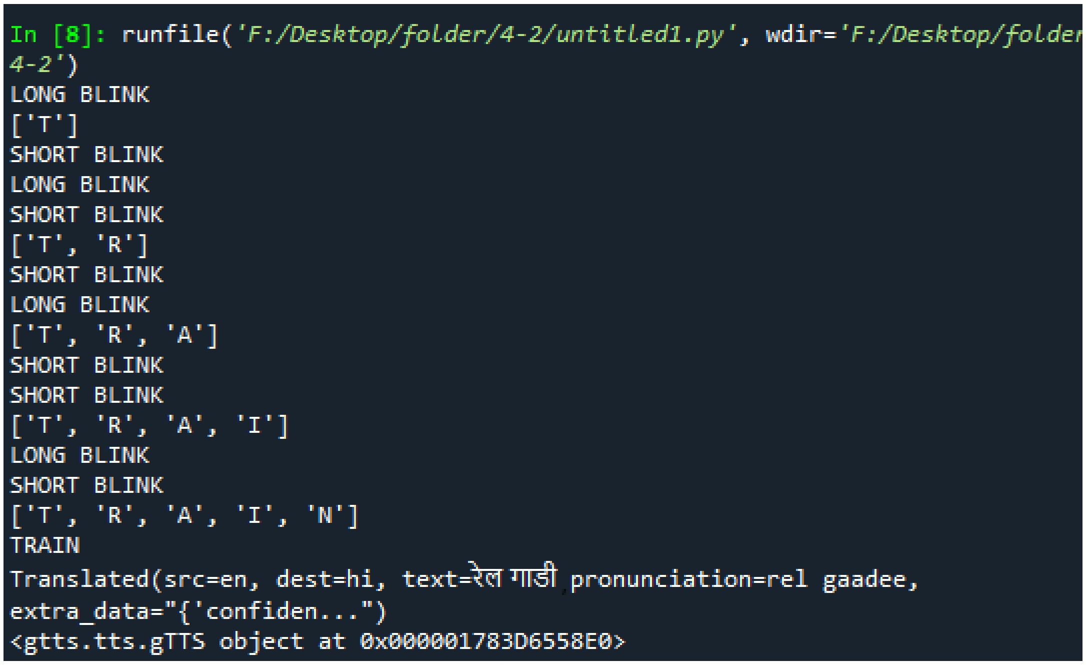

International Morse Code
1. The length of a dot is one unit.
2. A dash is three units.
3. The space between parts of the same letter is one unit.
4. The space between letters is three units.
5. The space between words is seven units.
A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S
T
U
V
W
X
Y
Z
1
2
3
4
5
6
7
8
9
0

Fig 3:Internatonal Morse code

IV.RESULT AND ANALYSIS

This program gives speech and text which is understandable by a normal human language, as chosen by user. Assuming the user selects Hindi as the language, the program will translate the word into Hindi text and speech and play the sound using the play sound library. If the word is not a valid one then it takes nearest possible word.



```
In [8]: runfile('F:/Desktop/folder/4-2/untitled1.py', wdir='F:/Desktop/folder
4-2')
LONG BLINK
['T']
SHORT BLINK
LONG BLINK
SHORT BLINK
['T', 'R']
SHORT BLINK
LONG BLINK
['T', 'R', 'A']
SHORT BLINK
SHORT BLINK
['T', 'R', 'A', 'I']
LONG BLINK
SHORT BLINK
['T', 'R', 'A', 'I', 'N']
TRAIN
Translated(src=en, dest=hi, text=रेल गाडी,pronunciation=rel gaadee,
extra_data="{'confiden...")
<gtts.tts.gTTS object at 0x000001783D6558E0>
```

Fig 4:Final result

V. CONCLUSIONS

It has been shown that the Eye Aspect ratio is effective at detecting blinks, compared to the traditional method of detecting blinks. For people who are paralyzed or disabled but still have some eye movement, Morse code blink detection can be very useful. Blink detection can be improved with the algorithms and techniques discussed. For these blinks, Morse code conversion is also in progress. Those who are unable to normally communicate may be able to use this method as a means of communicating with those who doesn’t have knowledge in Morse code.

REFERENCES

[1] Alternative Voice Communication Device using Eye Blink Detection for people with Speech Disorders Srividhya G, Murali S, A. Keerthana, Jaya Rubi

[2] BlazeFace: Sub-millisecond Neural Face Detection on Mobile GPUs Valentin Bazarevsky, Yury Kartynnik, Andrey Vakunov, Karthik Raveendran, Matthias Grundmann

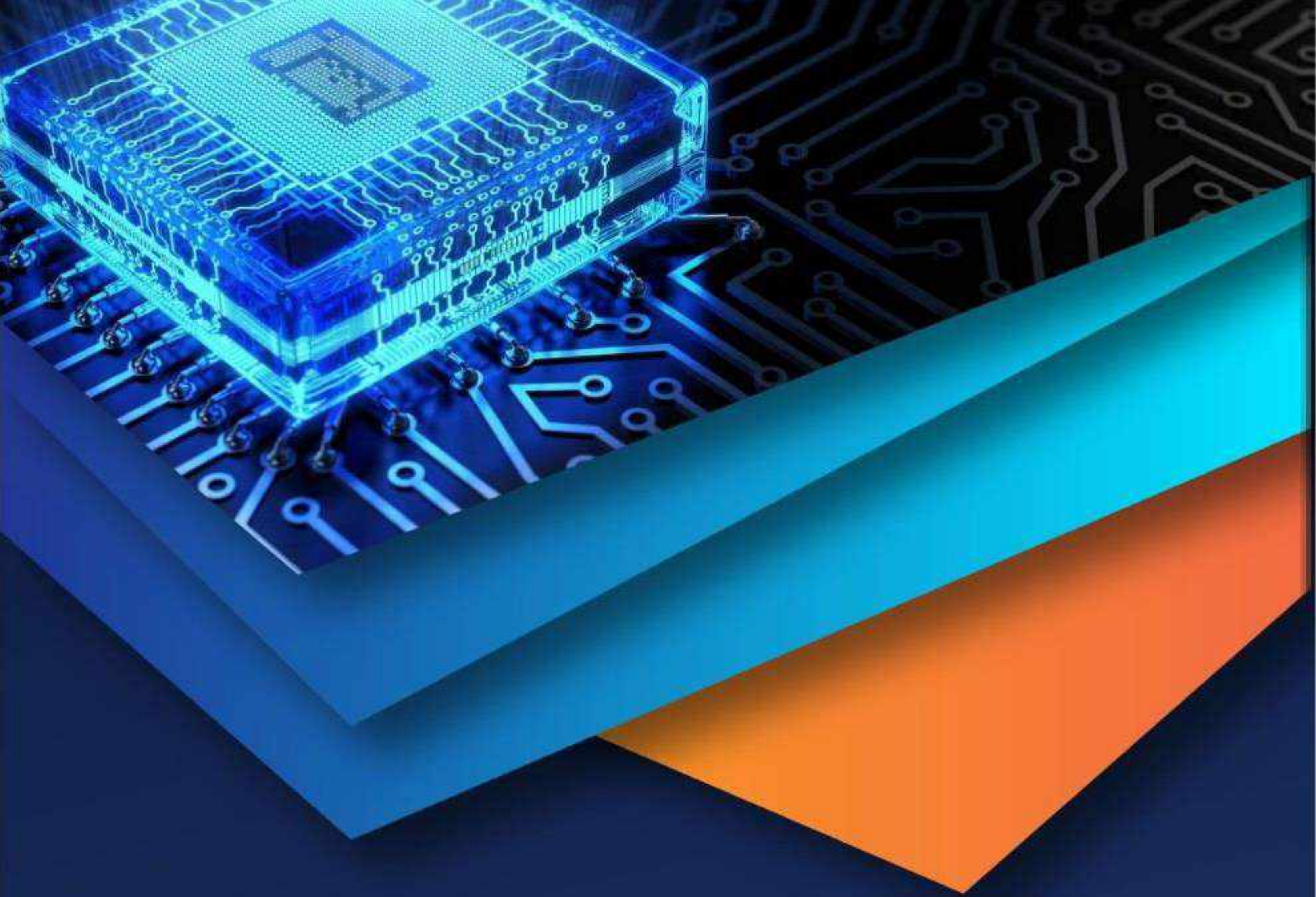
[3] Supervised Transformer Network for Efficient Face Detection Dong Chen, Gang Hua, Fang Wen, Jian Sun

[4] Attention Mesh: High-fidelity Face Mesh Prediction in Real-time I. Grishchenko, Artsiom Ablavatski, Y. Kartynnik, Karthik Raveendran, Matthias Grundmann

[5] Morse Codes Enter Using Finger Gesture Recognition Ricky Li, Minh Nguyen, Wei Qi Yan Department of Computer Science Auckland University of Technology, Auckland, 1010 New Zealand.

[6] Real-Time Eye Blink Detection using Facial Landmarks Tereza Soukupova and Jan ˇ Cech • Center for Machine Perception, Department of Cybernetics Faculty of Electrical Engineering, Czech Technical University in Prague

[7] A Novel Method for Eye Tracking and Blink Detection in video frames Leo Pauly, Deepa Sankar Division of Electronics and Communication Engineering School of Engineering Cochin University of Science and Technology



10.22214/IJRASET

INDEX COPERNICUS

45.98

ISRA JIF

IMPACT FACTOR: 7.129

TOGETHER WE REACH THE GOAL

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

ISSN No. : 2321-9653

iJRASET

International Journal for Research in Applied Science & Engineering Technology

IJRASET is indexed with Crossref for DOI-DOI : 10.22214

Website : www.ijraset.com, E-mail : ijraset@gmail.com

ISRA Journal Impact Factor: 7.429

45.98 INDEX COPERNICUS

THOMSON REUTERS
Researcher ID: N-9681-2016

10.22214/IJRASET

TOGETHER WE REACH THE GOAL
SJIF 7.429

Certificate

It is here by certified that the paper ID : IJRASET44536, entitled

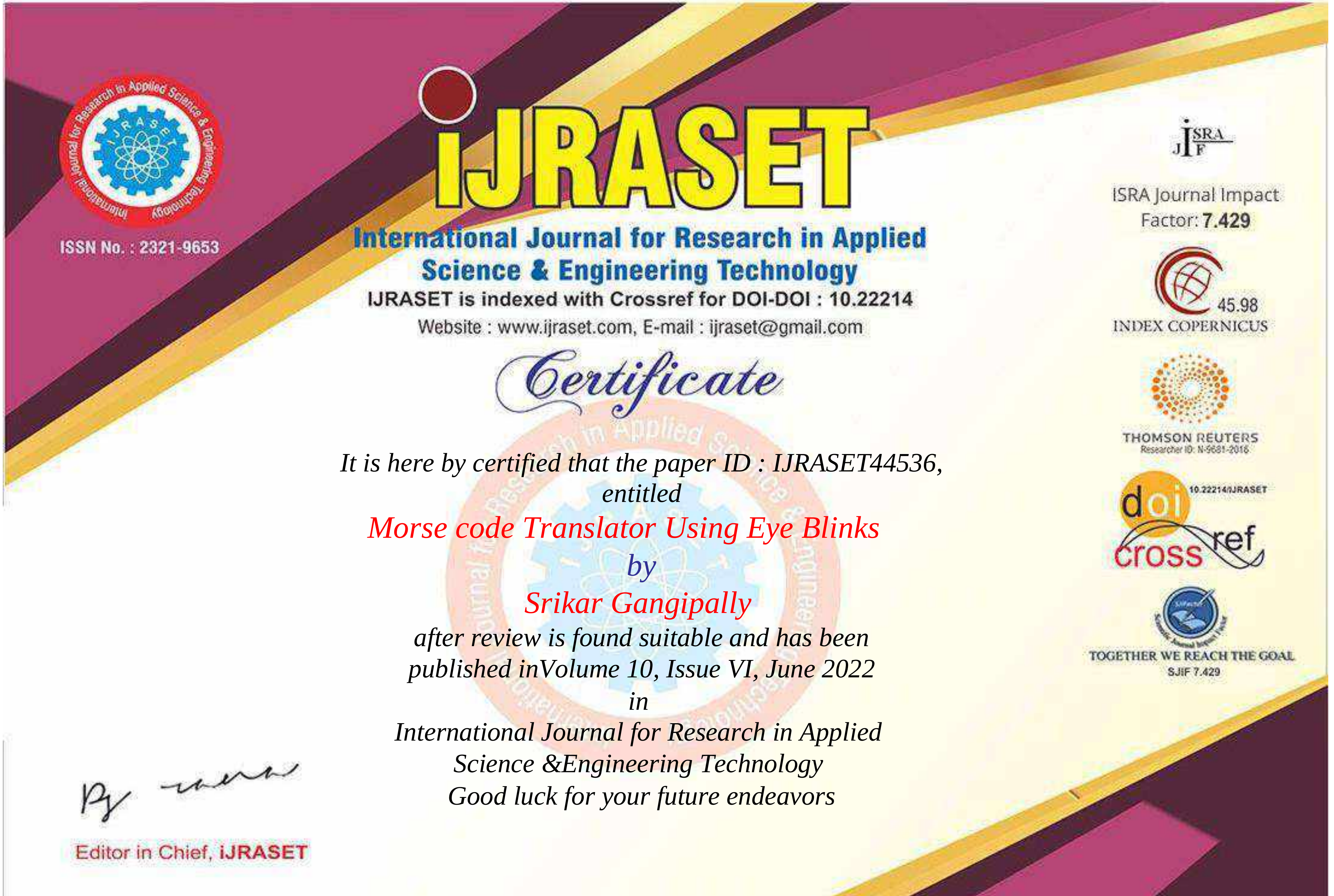
Morse code Translator Using Eye Blinks

by

Sainath Cheparthy

after review is found suitable and has been
published inVolume 10, Issue VI, June 2022
in
International Journal for Research in Applied
Science &Engineering Technology
Good luck for your future endeavors

Editor in Chief, IJRASET



ISSN No. : 2321-9653

IJRASET

International Journal for Research in Applied Science & Engineering Technology

IJRASET is indexed with Crossref for DOI-DOI : 10.22214

Website : www.ijraset.com, E-mail : ijraset@gmail.com

ISRA Journal Impact Factor: 7.429

45.98 INDEX COPERNICUS

THOMSON REUTERS Researcher ID: N-9681-2016

10.22214/IJRASET

TOGETHER WE REACH THE GOAL SJIF 7.429

Certificate

It is here by certified that the paper ID : IJRASET44536,
entitled
Morse code Translator Using Eye Blinks
by
Srikar Gangipally
after review is found suitable and has been
published inVolume 10, Issue VI, June 2022
in
International Journal for Research in Applied
Science &Engineering Technology
Good luck for your future endeavors

Editor in Chief, IJRASET

ISSN No. : 2321-9653

iJRASET

International Journal for Research in Applied Science & Engineering Technology

IJRASET is indexed with Crossref for DOI-DOI : 10.22214

Website : www.ijraset.com, E-mail : ijraset@gmail.com

ISRA Journal Impact Factor: 7.429

45.98 INDEX COPERNICUS

THOMSON REUTERS Researcher ID: N-9681-2016

10.22214/IJRASET

TOGETHER WE REACH THE GOAL

SJIF 7.429

Certificate

It is here by certified that the paper ID : IJRASET44536,
entitled
Morse code Translator Using Eye Blinks
by
Pranay Goud Iruvuri
after review is found suitable and has been
published inVolume 10, Issue VI, June 2022
in
International Journal for Research in Applied
Science &Engineering Technology
Good luck for your future endeavors

Editor in Chief, iJRASET