**ABSTRACT**

Sign language has played a vital and important role in today’s daily life. Communication between ability gaps has been high. The solution to the sign language problem is SIGN O. Sign language has filled the communication gap between the ones that can’t speak and use sign language to communicate with the world. And it also fills the void between ease of use of computer without much hardware interaction. Using the computer with the wave of a hand, using the hovering method to use the mouse and most importantly to use hand signs to convert signs to text for ease of interpretation between languages.

The solution is a program packed with python code, that helps in the process of interpretation bet ween the user and the system.

**INTRODUCTION**

SiGN-O is a desktop application that specializes in the conversion of sign language unit/symbols into words or letters of the English language. SiGN-O takes commands in real time from a live video being recorded and fed into the respective system. Then it processes the video cropping out the respective units from the data. And converts them into words copying them into a file for further use.

Using the latest Python Version for image processing to convert sign language symbols into text. And convert that text further into voice output that the user can understand and use freely.

**Product and Features**

The product is currently going to be a free for use app that any one can install and use in their PCs using the exe file itself. The product features a compatibility with Operating System Microsoft Windows 7 and above.

The product features the following services:

1. Conversion of Hand Signs into English Alphabets
2. Thresholding of hand signs to cropped shape.
3. Recording of video in real time.
4. Taking hand signs from the frames of video and cascading it into respective shapes and symbols.

The Product has the following components:

1. A web cam or any video input.
2. The cam should have more than 10 mega pixels for better thresholding and processing of image.
3. Python 2.7 for image processing.
4. OpenCV library for applying cascading algorithms and thresholding onto the frames of video.

**Software Requirements:**

1. Windows OS is most preferred
2. Python 2.7 IDLE + Interpreter
3. A video cam, web cam most preferred in Desktops or Laptops
4. OpenCV 2.0+

**Efficiency Requirements:**

1. RAM 2GB or more
2. More accurate Web Cam.
3. Using the application in a lighted room.
4. A dim lighted room might decrease accuracy.

**Detail Design:**

I was making the same app in C# before to understand the working and design of the app. Taking experience from some samples and understanding using C# gave me an outline about how the app can be made in python and how it shall work in real-time. With the knowledge and resources at hand. I have done my work in Python using both 3.7 and 2.7 at the initial stage. And Implementing OpenCV libraries to the mix to apply GAUSSIAN BLUR, THRESHHOLDING & CASCADING techniques to my code.

The final code right now implies Python 2.7 with OpenCV and uses the following libraries:

1. MATH
2. COPY
3. OPENCV v2.0
4. CV2
5. NUMPY

The code is designed to give the output in 4 formats right now:

1. Original Output
2. Applied Blur
3. Applied Threshold
4. Applied Contours

**Use Cases & Flow of Process:**

1.

|  |  |
| --- | --- |
| Name | Startup |
| ID | 001 |
| Description | The application starts with default values. |
| Actors | User, Application, Camera (WEBCAM) |
| Pre | Application is in running state and waiting for user input |
| Main Flow | The user must press B for Beginning the interpretation |
| Post | The program is ready to proceed to interpretation. |

2.

|  |  |
| --- | --- |
| Name | Getting and Processing Image |
| ID | 002 |
| Description | The application captures background image in frame |
| Actors | Application, Camera (WEBCAM), algorithm |
| Pre | The application has been started and is capturing video feed |
| Main Flow | The user must display his hand in the allotted frame to proceed to next step. |
| Post | The program now adds threshold and contours on the captured frame. |

3.

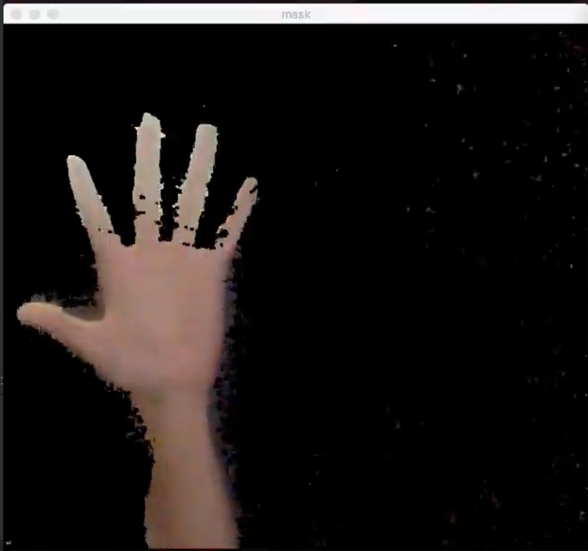
|  |  |
| --- | --- |
| Name | Cropping and Working on cropped image |
| ID | 003 |
| Description | The Application crops the image of a hand and applies threshold, blur and contours on the image/frame. |
| Actors | Application, Camera (WEBCAM), algorithm |
| Pre | The application has been started and is capturing video feed |
| Main Flow | The final image is created that has to be interpreted into words or commands. |
| Post | The program is now ready to read the hand symbols to convert to letters & Words. |

**Implementation Results:**

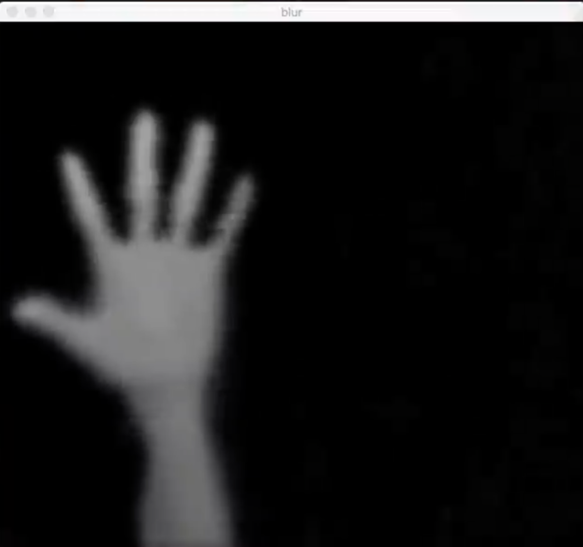
The project is now in the working stage. The program is successfully capturing data from a single frame from the video feed. The program applies threshold, blur and contours which display a compete and clear hand in the final window.

The achieved stage shows 4 windows which display the steps that the image took to reach the final stage.

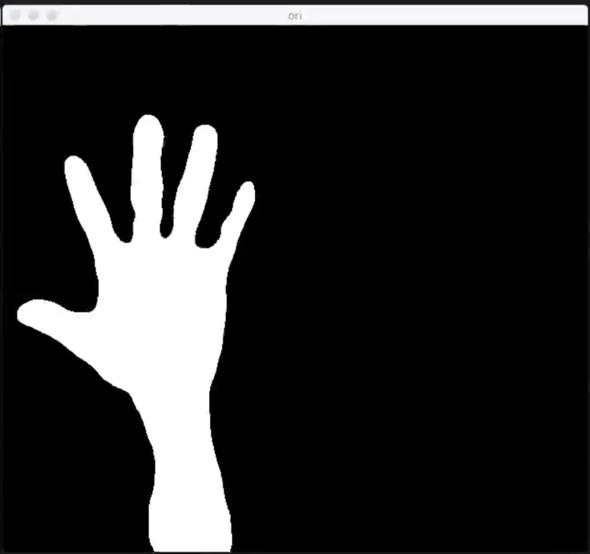
1. Cropping Image



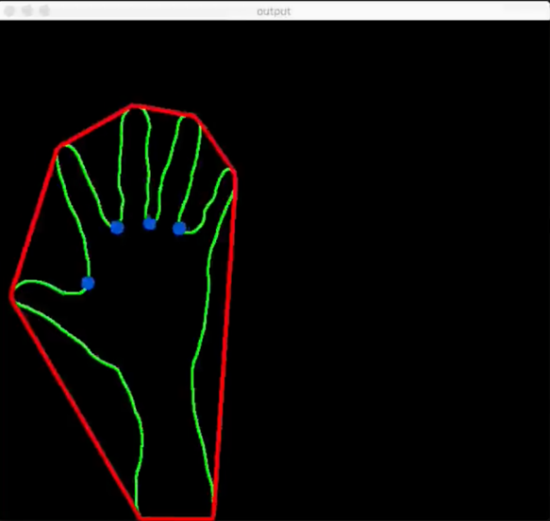
1. Applying Blur



1. Blotting on Blury Image



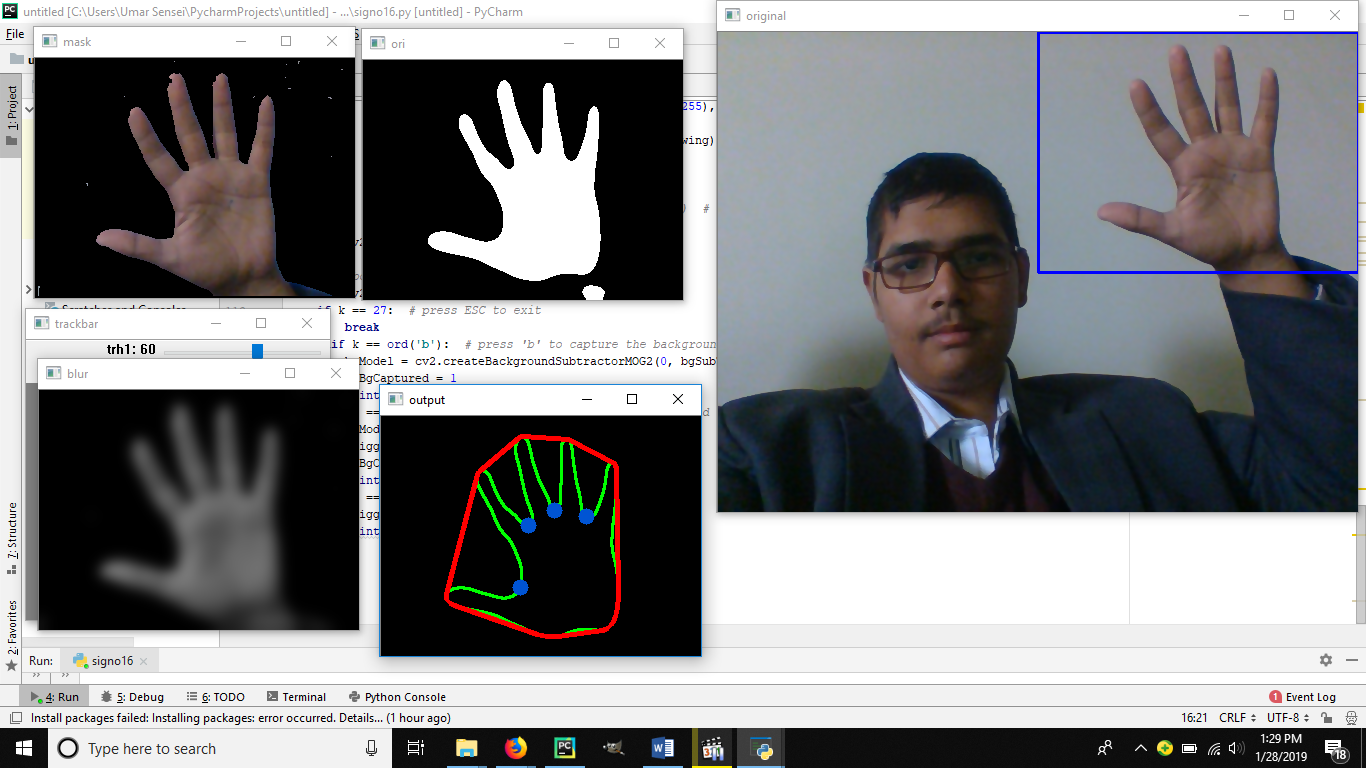
1. Aplying Contours



1. Original Image



**^@^ SAMPLE OUTPUT ON NEXT PAGE ^@^**



**Future Work:**

The program will be able to fulfill the following services:

1. Convert hand signs into words
2. Convert hand signs into letters
3. Convert hand signs into commands
4. More accurate capturing of video.
5. Better thresholding and blotting mechanisms
6. Adding of the module that converts the result text to voice output.
7. Addition of a GUI to the program for a friendly user interface.

Final Year Project (FYP)

Project Report

Title: SIGN O: The Sign Language Operator

Creator: Umar Safdar

Roll No: #153161

Dept: BSCS

Supervisor: Dr. Kaleem Razzaq Malik

AIR UNIVERSITY

MULTAN

2019



