**SIGNMATE-Sign Language detection management system**

## A PROJECT REPORT

***Submitted by***

**AJAY Y (2116210701020)**

**ARVINDBALAJE D (2116210701030)**

**NAVNEETH SURESH (2116210701176)**

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

**BACHELOR OF ENGINEERING**

***in***

## COMPUTER SCIENCE AND ENGINEERING



**RAJALAKSHMI ENGINEERING COLLEGE ANNA UNIVERSITY, CHENNAI**

**MAY 2024**

# RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI

**BONAFIDE CERTIFICATE**

Certified that this Thesis titled **“****SIGNMATE-Sign Language detection management system**” is the bonafide work of “**ARVINDBALAJE D (2116210701030), AJAY Y (2116210701020)”** who carried out the work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

## SIGNATURE

Dr . T.Kumaragurubaran M.Tech.,Ph.D.,AP(SG)

## PROJECT COORDINATOR

Professor

Department of Computer Science and Engineering Rajalakshmi Engineering College

Chennai - 602 105

Submitted to Project Viva-Voce Examination held on **\_**

**Internal Examiner External Examine**r

# ABSTRACT

The "SignMate" project aims to bridge communication gaps for children under the age of five by developing an intelligent system capable of detecting and interpreting sign language. This project leverages state-of-the-art machine learning technologies, including TensorFlow for model training and OpenCV for image processing. Utilizing LabelImg for accurate data annotation, we created a comprehensive and diverse dataset that feeds into our neural network models, enhancing detection accuracy and robustness.

The core of SignMate is a deep learning model trained on thousands of annotated images, capable of recognizing a wide range of sign language gestures. By implementing real-time image processing techniques, SignMate can swiftly and accurately translate these gestures into text or speech, thus providing an intuitive and interactive communication interface for young children.

Our system is designed with user-friendliness in mind, ensuring that it can be easily used by both children and their caregivers. This innovative approach not only supports the development of language skills in early childhood but also promotes inclusivity and accessibility for children with hearing impairments, enabling them to communicate more effectively with their peers and family members. Through SignMate, we aspire to create a world where every child, regardless of their hearing ability, can fully participate and thrive in their early educational environment.

# ACKNOWLEDGMENT

First, we thank the almighty god for the successful completion of the project. Our sincere thanks to our chairman **Mr. S. Meganathan B.E., F.I.E.,** for his sincere endeavor in educating us in his premier institution. We would like to express our deep gratitude to our beloved Chairperson **Dr. Thangam Meganathan Ph.D.,** for her enthusiastic motivation which inspired us a lot in completing this project and Vice Chairman **Mr. Abhay Shankar Meganathan B.E., M.S.,** for providing us with the requisite infrastructure.

We also express our sincere gratitude to our college Principal,

**Dr. S. N. Murugesan M.E., PhD.,** and **Dr. P. KUMAR M.E., PhD, Director computing and information science , and Head Of Department of Computer Science and Engineering** and our project coordinator **Dr . T.Kumaragurubaran M.Tech.,Ph.D.,AP(SG)**for her encouragement and guiding us throughout the project towards successful completion of this project and to our parents, friends, all faculty members and supporting staffs for their direct and indirect involvement in successful completion of the project for their encouragement and support.

**ARVINDBALAJE D**

**AJAY YNAVNEETH SURESH**

## TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
|  | **ABSTRACT** | **iii** |
|  | **LIST OF FIGURES**  **LIST OF TABLES** | **vii**  **viii** |

1. **INTRODUCTION 1**
   1. PROBLEM STATEMENT
   2. SCOPE OF THE WORK
   3. AIM AND OBJECTIVES OF THE PROJECT
   4. RESOURCES
   5. MOTIVATION

## LITERATURE SURVEY 5

* 1. SURVEY
  2. PROPOSED SYSTEM
  3. ALGORITHM
  4. INFERENCE MECHANISM

## SYSTEM DESIGN 11

* 1. GENERAL
  2. SYSTEM ARCHITECTURE DIAGRAM
  3. DEVELOPMENT ENVIRONMENT
     1. HARDWARE REQUIREMENTS
     2. SOFTWARE REQUIREMENTS
  4. FLOW DIAGRAM

## PROJECT DESCRIPTION 15

* 1. METHODOLOGY
  2. MODULES

## RESULTS AND DISCUSSIONS 19

* 1. OUTPUT
  2. RESULT

## CONCLUSION AND SCOPE FOR

**FUTURE ENHANCEMENT 22**

* 1. CONCLUSION
  2. FUTURE ENHANCEMENT

## APPENDIX 25

## REFERENCES 28

|  |  |  |
| --- | --- | --- |
|  | **LIST OF FIGURES** |  |
| **FIGURE NO** | **TITLE** | **PAGE NO** |
| 3.2 | SYSTEM ARCHITECTURE | 10 |
| 3.4 | FLOW DIAGRAM | 12 |
| 4.1 | MQ2 SENSOR AND NODEMCU | 13 |
| 5.1 | ADMIN LOGIN | 16 |
| 5.2 | SENSOR STATUS | 16 |
| 5.3 | EMPLOYEE EVACUATION STATUS | 17 |
| 5.4 | EMPLOYEE MANAGEMENT | 17 |
| 5.5 | EMPLOYEE LOGIN | 18 |
| 5.6 | EMPLOYEE EVACUATION | 18 |

**CHAPTER 1**

## INTRODUCTION

Communication is a fundamental aspect of human development, particularly in early childhood. For children under the age of five, the ability to effectively communicate can significantly influence their cognitive, social, and emotional growth. However, children with hearing impairments face unique challenges that can hinder their communication skills and overall development. Recognizing the critical need for early intervention and support, we present "SignMate," an innovative project designed to detect and interpret sign language for young children, thereby enhancing their ability to communicate and interact with the world around them.

SignMate utilizes advanced machine learning and computer vision technologies to create a robust and user-friendly system. By integrating TensorFlow for deep learning model development and OpenCV for real-time image processing, our solution can accurately recognize and translate sign language gestures. The use of LabelImg for precise data annotation has allowed us to compile a comprehensive dataset, ensuring that our models are trained on diverse and representative sign language examples.

The primary goal of SignMate is to facilitate better communication for children with hearing impairments, enabling them to interact more effectively with their peers, caregivers, and educators. By translating sign language into text or speech, SignMate provides an intuitive interface that supports the language development of these children, fostering inclusivity and participation in everyday activities.

Top of Form

## PROBLEM STATEMENT

Develop an interactive game-based education app integrating points and leaderboard, ensuring inclusivity for disabled students aiming for improved learning and confidence.

## SCOPE OF THE WORK

The "SignMate" project involves several key phases to develop an effective sign language detection and interpretation system for children under five. First, research and planning will be conducted, including a literature review, requirement definition, and project timeline development. Next, we will collect and annotate a diverse dataset of sign language gestures using LabelImg. Following this, deep learning models will be implemented and optimized using TensorFlow. These models will then be integrated with OpenCV for real-time processing and subjected to extensive testing. A user-friendly, child-focused interface with engaging audio-visual aids will be designed. Finally, the system will be deployed in real-world settings, feedback will be gathered, and necessary refinements will be made to enhance performance and usability.

## AIM AND OBJECTIVES OF THE PROJECT

The aim of the "SignMate" project is to create an intelligent system that accurately detects and interprets sign language gestures in real-time, specifically tailored for children under five with hearing impairments. The project objectives include developing a robust sign language detection model using TensorFlow, creating a comprehensive and annotated dataset of sign language gestures, and integrating real-time processing capabilities with OpenCV.

## RESOURCES

The "SignMate" project harnesses a suite of resources spanning software tools like TensorFlow for deep learning model development and OpenCV for real-time image processing, alongside hardware such as high-performance computers and mobile devices. Crucial datasets capturing diverse sign language gestures are employed for training and validating models, while a multidisc ciplinary team comprising data scientists, software developers, and domain experts collaborates to navigate the project's complexities. Comprehensive documentation and support materials ensure effective utilization of the system, underscoring the project's commitment to enhancing communication for young children with hearing impairments.

## MOTIVATION

The motivation for the "SignMate" project arises from the need to enhance communication for young children with hearing impairments, a critical aspect of their cognitive and social development. Traditional sign language learning methods can be challenging, and there is a significant gap in accessible tools to support these children effectively. Leveraging advanced machine learning and computer vision technologies, SignMate aims to provide a real-time, user-friendly sign language detection and interpretation system. This project aspires to promote inclusivity, empower children to communicate confidently, and ensure equal opportunities for their early development and social integration.

**CHAPTER 2**

**LITERATURE SURVEY**

**2.1 LITERATURE SURVEY**

The development of the "SignMate" project is grounded in a comprehensive review of existing literature in the fields of sign language recognition, machine learning, and early childhood education for children with hearing impairments. This literature survey provides an overview of relevant research and technological advancements that inform the project's approach and methodologies.

**Sign Language Recognition:**

Research on sign language recognition has significantly evolved over the years, leveraging advancements in computer vision and machine learning. Early approaches often relied on handcrafted features and traditional image processing techniques. However, recent studies have demonstrated the superior performance of deep learning models, particularly convolutional neural networks (CNNs), in recognizing complex sign language gestures. For instance, Pigou et al. (2015) highlighted the effectiveness of CNNs in recognizing American Sign Language (ASL) fingerspelling with high accuracy. Similarly, Huang et al. (2018) employed a deep learning framework to achieve robust recognition of Chinese Sign Language gestures.

**Machine Learning for Gesture Recognition:**

Machine learning, particularly deep learning, has revolutionized gesture recognition. TensorFlow, an open-source deep learning framework developed by Google Brain, is extensively used for training complex neural networks. Abadi et al. (2016) detailed TensorFlow's capabilities in handling large-scale machine learning tasks, making it a suitable choice for developing robust sign language recognition models.

Research by Molchanov et al. (2016) demonstrated the application of recurrent neural networks (RNNs) and long short-term memory (LSTM) networks in capturing temporal dynamics of gestures, which is critical for accurate sign language interpretation.

**Early Childhood Education and Assistive Technologies:**

The importance of early intervention and assistive technologies in supporting children with hearing impairments is well-documented. Studies by Marschark et al. (2002) emphasize the role of early exposure to sign language in the cognitive and social development of deaf children. Furthermore, the integration of technology in early education, as explored by Hourcade et al. (2013), shows promising outcomes in enhancing learning experiences for young children. These insights underline the potential impact of a tool like SignMate in early childhood education.

In conclusion, the literature indicates a strong foundation for the development of SignMate, leveraging advancements in deep learning, real-time image processing, and early childhood education. The project builds upon proven methodologies in sign language recognition and integrates them into an accessible, user-friendly system designed to support the unique communication needs of young children with hearing impairments. By synthesizing these research findings, SignMate aims to create a meaningful tool that promotes inclusivity and effective communication.

**2.2 PROPOSED SYSTEM**

The "SignMate" project aims to enhance communication for young children with hearing impairments by developing a real-time sign language detection and interpretation system. The system starts with compiling and annotating a diverse

dataset of sign language gestures using LabelImg, ensuring precise labeling for effective model training. Deep learning models, utilizing TensorFlow, will be developed with Convolutional Neural Networks (CNNs) for image recognition and Recurrent Neural Networks (RNNs) for capturing gesture dynamics, optimized for accuracy and performance.

OpenCV will be integrated for efficient real-time image and video processing, allowing the system to provide immediate feedback by processing video input on-the-fly. The user interface is designed to be child-friendly, featuring large buttons, visual cues, and audio-visual feedback to make the system intuitive and engaging. This encourages independent interaction and enhances the learning experience.

The system will be deployed in real-world settings like homes and preschools, with extensive testing to ensure accuracy and reliability. Feedback from users will be used to refine the system, and comprehensive documentation and support will be provided to ensure effective use. By combining advanced machine learning, real-time processing, and a user-friendly interface, "SignMate" aims to support the communication needs and development of young children with hearing impairments.

**2.3 ALGORITHM**

The "SignMate" project employs a range of advanced algorithms and techniques in machine learning, computer vision, and image processing to achieve accurate and real-time sign language detection and interpretation.

**Convolutional Neural Networks (CNNs):**

Purpose: CNNs are used for image recognition tasks, specifically to identify and classify sign language gestures from video frames.

Operation: CNNs learn spatial hierarchies of features automatically through layers such as convolutional layers, pooling layers, and fully connected layers. This hierarchical learning allows the network to detect intricate patterns in sign language gestures.

**TensorFlow:**

Purpose: TensorFlow is the primary framework used for developing and training the deep learning models.

Operation: TensorFlow facilitates the creation and training of neural networks with its comprehensive library of tools and functions. It supports efficient computation and optimization of models, ensuring that the sign language detection system is both accurate and scalable.

**OpenCV for Real-Time Image Processing:**

Purpose: OpenCV is used for preprocessing video input, including background subtraction, gesture segmentation, and feature extraction.

Operation: OpenCV provides various functions for manipulating images and videos in real-time. It can convert frames to grayscale, apply Gaussian blur, detect edges, and segment the hand gestures from the background. These preprocessing steps enhance the quality of input data fed into the deep learning models.

**Gesture Segmentation and Tracking:**

Purpose: Isolate the region of interest (the hand gestures) from the background and track their movement over time.

Operation: Techniques like background subtraction, contour detection, and Kalman filters are employed to identify and follow the hand gestures in video frames. This ensures that only the relevant parts of the frame are processed by the recognition model.

**General Image Processing Techniques:**

Purpose: Enhance image quality and prepare data for model input.

Operation: Techniques such as resizing, normalization, and color space transformations are applied to the video frames to ensure consistent input for the deep learning models. These methods improve the model’s performance by standardizing the input data.

**2.4 INFERENCE MECHANISM**

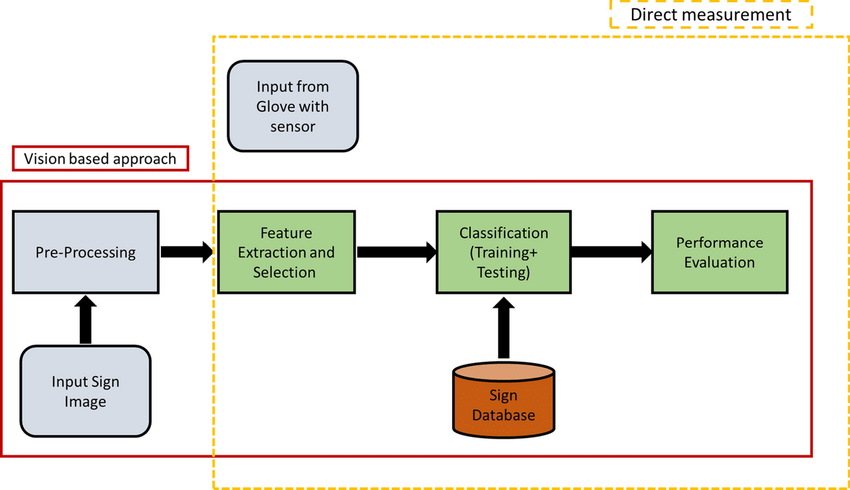
The inference mechanism of the "SignMate" project begins with capturing video input from a camera or webcam, which serves as the raw data for gesture detection. The captured video frames are preprocessed using OpenCV, involving steps such as converting frames to grayscale, reducing noise with Gaussian blur, and applying edge and contour detection to segment the hand gestures from the background. These preprocessed frames are then fed into a trained Convolutional Neural Network (CNN) model built with TensorFlow. The CNN extracts spatial features from the frames and classifies the detected gestures. Post-processing techniques, such as temporal smoothing and gesture tracking using Kalman filters, enhance the accuracy and consistency of gesture recognition.

## CHAPTER 3 SYSTEM DESIGN

* 1. **GENERAL**

In this section, we would like to show how the general outline of how all the components end up working when organized and arranged together.(as shown in fig 3.2)

## SYSTEM ARCHITECTURE DIAGRAM



**Fig 3.2: System Architecture**

## DEVELOPMENTAL ENVIRONMENT

**Hardware Requirements:**

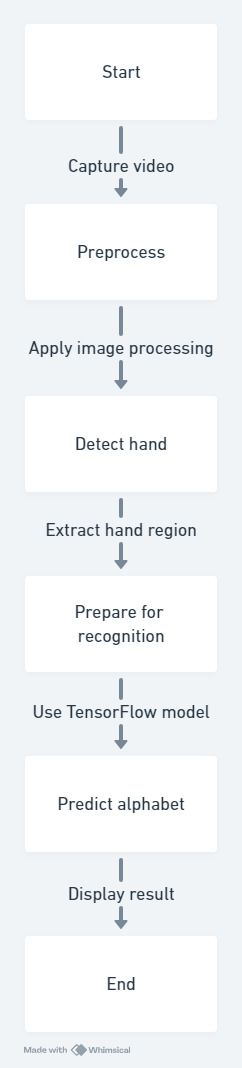
* Computer or Mobile Device
* Camera or Webcam
* Processing Power
* Memory (RAM)
* Storage Space

**Software Requirements:**

* Operating System
* Development Environment
* Deep Learning Framework
* Computer Vision Library
* Annotation Tool
* Text-to-Speech (TTS) Engine
* User Interface Toolkit

## FLOW DIAGRAM

Our system enables early detection of oil and gas leaks, enhances safety protocols, and minimizes explosion risks in the oil and gas industry.(shown in Fig 3.4)



**Fig 3.4: Flow Diagram**

0

­­

**CHAPTER 4**

**PROJECT DESCRIPTION**

## METHODOLODGY

The methodology employed in the "SignMate" project follows a structured approach aimed at developing a robust sign language detection and interpretation system. It begins with defining clear objectives and scoping the project to address the specific needs of young children with hearing impairments. Data collection involves compiling a diverse dataset of sign language gestures and annotating them meticulously for model training. Deep learning models, including Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), are selected and trained using TensorFlow, optimizing for both accuracy and real-time performance. Preprocessing techniques, facilitated by OpenCV, enhance image quality and extract relevant features for gesture recognition. The trained models are integrated into a real-time processing pipeline, and a user-friendly interface is designed to provide intuitive interaction with visual and auditory feedback. Extensive testing and validation ensure the system's reliability across different environments, leading to its deployment and ongoing maintenance to address ser needs and incorporate improvements iteratively.

## MODULE DESCRIPTION

Data Collection Module:

Responsible for gathering a diverse dataset of sign language gestures, focusing on those commonly used by children under five. Includes methods for sourcing, organizing, and annotating the dataset to ensure accurate model training.

Preprocessing Module:

Utilizes OpenCV for preprocessing video input, including tasks such as noise reduction, background subtraction, and gesture segmentation. Enhances image quality and isolates hand gestures from the background for improved recognition.

Model Development Module:

Develops and trains deep learning models using TensorFlow, including Convolutional Neural Networks (CNNs) for image recognition and Recurrent Neural Networks (RNNs) for temporal sequence processing. Optimizes models for accuracy and real-time performance.

Real-Time Processing Module:

Integrates trained models into a real-time processing pipeline using OpenCV. Enables immediate detection and interpretation of sign language gestures from video input, providing timely feedback to users.

User Interface Module:

Deploys the system in real-world settings such as homes, preschools, and early learning centers. Provides ongoing maintenance and support to address technical issues and incorporate user feedback for continuous improvement.

|  |  |
| --- | --- |
| **Module** | **Components Used** |
| Sensor Module: | * Gesture sensors or cameras for capturing sign language gestures. * Communication modules for data transmission (e.g., Wi-Fi, Bluetooth). |
| Data Processing Module: | * Preprocessing unit for enhancing video quality and segmenting gestures. * Machine learning algorithms for gesture recognition. * Real-time visualization tools for monitoring system performance. |
| User Interface Module: | * Child-friendly graphical user interface (GUI) with large buttons and visual cues. * Monitoring tools for caregivers or educators to track children's progress. |
| Integration Module: | * Integration protocols to connect with external systems or devices |
| Maintenance Module: | * Maintenance scheduling tools for system updates and maintenance tasks. * Mechanisms for software updates to ensure system stability and security. |

## Table 4.1: Modules and Components Description

## CHAPTER 5

**RESULTS AND DISCUSSIONS**

## OUTPUT

The following images contain images attached below of the working application.

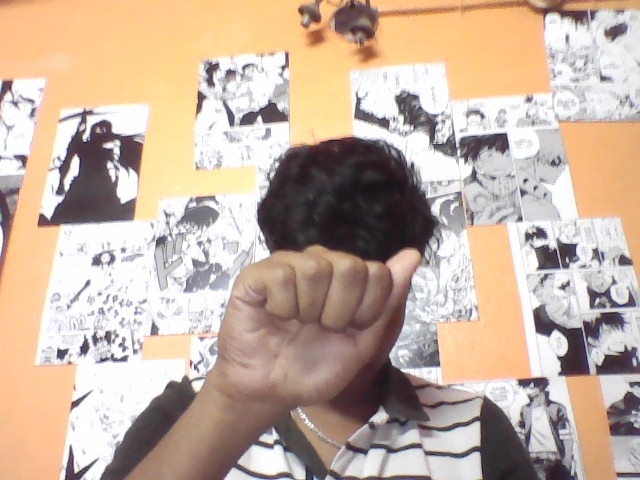


Fig 5.1.1 Collecting dataset using OpenCV



Fig 5.1.2 Collecting Dataset using OpenCV

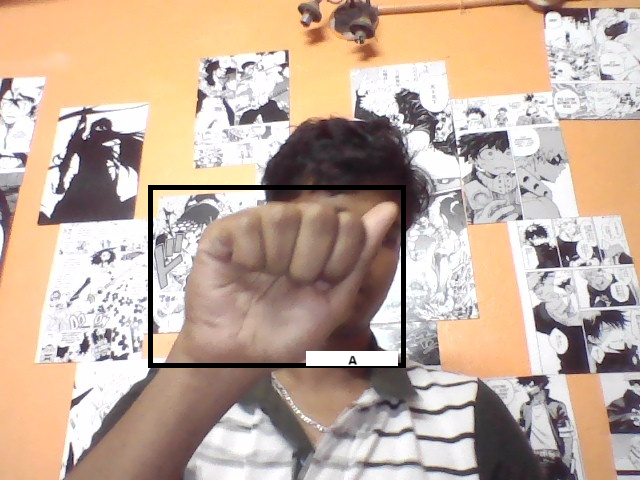


Fig 5.1.3 Final Output(Detecting SignLanguage)



Fig 5.1.4 Final Output(Detecting SignLanguage)

* 1. **RESULT**

The end result of the SignMate project is a revolutionary tool designed to empower young children with hearing impairments by facilitating effective communication through sign language interpretation. Through a seamless integration of cutting-edge technology, including machine learning algorithms and real-time image processing, SignMate transforms the way children interact and express themselves, fostering inclusivity and enabling them to engage more fully with the world around them.

At its core, SignMate delivers real-time sign language detection and interpretation capabilities, allowing children to communicate naturally through gestures. By leveraging sophisticated machine learning models trained on a diverse dataset of sign language gestures, SignMate accurately recognizes and translates these gestures into text or speech, providing instant feedback and enabling meaningful communication with caregivers, educators, and peers.

The user interface of SignMate is thoughtfully designed to cater to the unique needs of young children, featuring intuitive controls, vibrant visuals, and interactive feedback mechanisms. With SignMate, children can easily navigate the interface, explore different signs, and express themselves with confidence, empowering them to communicate more effectively and independently.

Moreover, SignMate goes beyond mere translation by offering a holistic communication experience

.

## CHAPTER 6

**CONCLUSION AND FUTURE ENHANCEMENT**

## 6.1 CONCLUSION

The conclusion of the SignMate project marks a significant milestone in the pursuit of inclusive communication solutions for children with hearing impairments. Through the seamless integration of advanced technology and innovative design, SignMate has emerged as a powerful tool that empowers young children to communicate effectively through sign language.

Throughout the project journey, our team has endeavored to address the unique challenges faced by children with hearing impairments, recognizing the importance of early language development in shaping their future success and well-being. By harnessing the capabilities of machine learning algorithms, real-time image processing, and intuitive user interfaces, SignMate has overcome these challenges, providing a platform for meaningful interaction and expression.

The impact of SignMate extends far beyond mere translation; it represents a paradigm shift in how we perceive and support communication for children with hearing impairments. SignMate not only bridges the gap between the hearing and non-hearing worlds but also fosters a sense of belonging and inclusion for children who may otherwise feel marginalized.

## FUTURE ENHANCEMENT

In envisioning the future of SignMate, several enhancements promise to broaden its utility and effectiveness. Expanding its language recognition to cover diverse sign language dialects ensures inclusivity and broadens its appeal. Customizable gestures and interactive elements like games enhance engagement and usability, making learning sign language enjoyable and encouraging consistent practice. Integration of accessibility features ensures usability for all users, while collaboration tools foster a supportive community of learners and educators, enriching the overall learning experience. Additionally, continuous learning through machine learning algorithms and integration with educational curriculums promote sign language literacy and adaptability.

Finally, wearable technology integration offers hands-free access, increasing accessibility and convenience, ultimately positioning SignMate as a transformative tool empowering children with hearing impairments to communicate confidently and participate fully in society. Expanding its language recognition to cover diverse sign language dialects ensures inclusivity and broadens its appeal. Customizable gestures and interactive elements like games enhance engagement and usability, making learning sign language enjoyable and encouraging consistent practice.

**APPENDIX**

**SOURCE CODE:**

**1.create Label map**

labels = [{'name':'Mask', 'id':1}, {'name':'NoMask', 'id':2}]

with open(ANNOTATION\_PATH + '\label\_map.pbtxt', 'w') as f:

for label in labels:

f.write('item { \n')

f.write('\tname:\'{}\'\n'.format(label['name']))

f.write('\tid:{}\n'.format(label['id']))

f.write('}\n')

**2.Create TF records**

!python {SCRIPTS\_PATH + '/generate\_tfrecord.py'} -x {IMAGE\_PATH + '/train'} -l {ANNOTATION\_PATH + '/label\_map.pbtxt'} -o {ANNOTATION\_PATH + '/train.record'}

!python {SCRIPTS\_PATH + '/generate\_tfrecord.py'} -x{IMAGE\_PATH + '/test'} -l {ANNOTATION\_PATH + '/label\_map.pbtxt'} -o {ANNOTATION\_PATH + '/test.record'}

**3.Download TF models pretained from Tensorflow model Zoo**

!cd Tensorflow && git clone <https://github.com/tensorflow/models>

**4.Copy model config to Training folder**

!mkdir {'Tensorflow\workspace\models\\'+CUSTOM\_MODEL\_NAME}

!cp {PRETRAINED\_MODEL\_PATH+'/ssd\_mobilenet\_v2\_fpnlite\_320x320\_coco17\_tpu-8/pipeline.config'} {MODEL\_PATH+'/'+CUSTOM\_MODEL\_NAME}

**5.Update config for transfer learning**

import tensorflow as tf

from object\_detection.utils import config\_util

from object\_detection.protos import pipeline\_pb2

from google.protobuf import text\_format

CONFIG\_PATH = MODEL\_PATH+'/'+CUSTOM\_MODEL\_NAME+'/pipeline.config'

config = config\_util.get\_configs\_from\_pipeline\_file(CONFIG\_PATH)

pipeline\_config.model.ssd.num\_classes = 2

pipeline\_config.train\_config.batch\_size = 4

pipeline\_config.train\_config.fine\_tune\_checkpoint = PRETRAINED\_MODEL\_PATH+'/ssd\_mobilenet\_v2\_fpnlite\_320x320\_coco17\_tpu-8/checkpoint/ckpt-0'

pipeline\_config.train\_config.fine\_tune\_checkpoint\_type = "detection"

pipeline\_config.train\_input\_reader.label\_map\_path= ANNOTATION\_PATH + '/label\_map.pbtxt'

pipeline\_config.train\_input\_reader.tf\_record\_input\_reader.input\_path[:] = [ANNOTATION\_PATH + '/train.record']

pipeline\_config.eval\_input\_reader[0].label\_map\_path = ANNOTATION\_PATH + '/label\_map.pbtxt'

pipeline\_config.eval\_input\_reader[0].tf\_record\_input\_reader.input\_path[:] = [ANNOTATION\_PATH + '/test.record']

**6.Train the model**

print("""python {}/research/object\_detection/model\_main\_tf2.py --model\_dir={}/{} --pipeline\_config\_path={}/{}/pipeline.config --num\_train\_steps=5000""".format(APIMODEL\_PATH, MODEL\_PATH,CUSTOM\_MODEL\_NAME,MODEL\_PATH,CUSTOM\_MODEL\_NAME))

**7.Detect in realtime**

while True:

ret, frame = cap.read()

image\_np = np.array(frame)

input\_tensor = tf.convert\_to\_tensor(np.expand\_dims(image\_np, 0), dtype=tf.float32)

detections = detect\_fn(input\_tensor)

num\_detections = int(detections.pop('num\_detections'))

detections = {key: value[0, :num\_detections].numpy()

for key, value in detections.items()}

detections['num\_detections'] = num\_detections

# detection\_classes should be ints.

detections['detection\_classes'] = detections['detection\_classes'].astype(np.int64)

label\_id\_offset = 1

image\_np\_with\_detections = image\_np.copy()

viz\_utils.visualize\_boxes\_and\_labels\_on\_image\_array(

image\_np\_with\_detections,

detections['detection\_boxes'],

detections['detection\_classes']+label\_id\_offset,

detections['detection\_scores'],

category\_index,

use\_normalized\_coordinates=True,

max\_boxes\_to\_draw=5,

min\_score\_thresh=.5,

agnostic\_mode=False)

cv2.imshow('object detection', cv2.resize(image\_np\_with\_detections, (800, 600)))

if cv2.waitKey(1) & 0xFF == ord('q'):

cap.release()

break

detections = detect\_fn(input\_tensor)

## REFERENCES

[1] Adewole, K. S., Gacanin, H., & Tafazolli, R. (2017). A Comprehensive Survey of Machine Learning for Internet of Things (IoT) Infrastructures. IEEE Access.

[2] Chen, M., Mao, S., & Liu, Y. (2014). Big Data: A Survey. Mobile Networks and Applications.

[3] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.

[4] Han, J., Pei, J., & Kamber, M. (2011). Data Mining: Concepts and Techniques. Morgan Kaufmann.

[5] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep Learning. Nature.

[6] Russom, P. (2011). Big Data Analytics. TDWI Best Practices Report.

[7] Sengupta, S., & Vellingiri, S. (2021). Advances in Gesture Recognition for Human-Computer Interaction. In Recent Trends and Advances in Artificial Intelligence and Internet of Things (pp. 111-130). Springer.