INTERNSHIP REPORT

Gayatri Samal
20UAI057
TY- CSE-AI
Samalg2502@gmail.com

Abstract

This report is being presented as part of Internship building course to describe the details of the internship done as part of the winter internship process of the DKTE Curriculum.

Introduction

As a part of the IBM Skill Build Internship program from AICTE portal, I was given the role of AI/ML developer in the project of building a Sign Language Recognition system. This internship was a valuable learning experience that helped me understand the practical aspects of machine learning and artificial intelligence, as well as the corporate work environment. This report outlines my experience and insights gained from the project.

Internship At
IBM Skill Build - AICTE

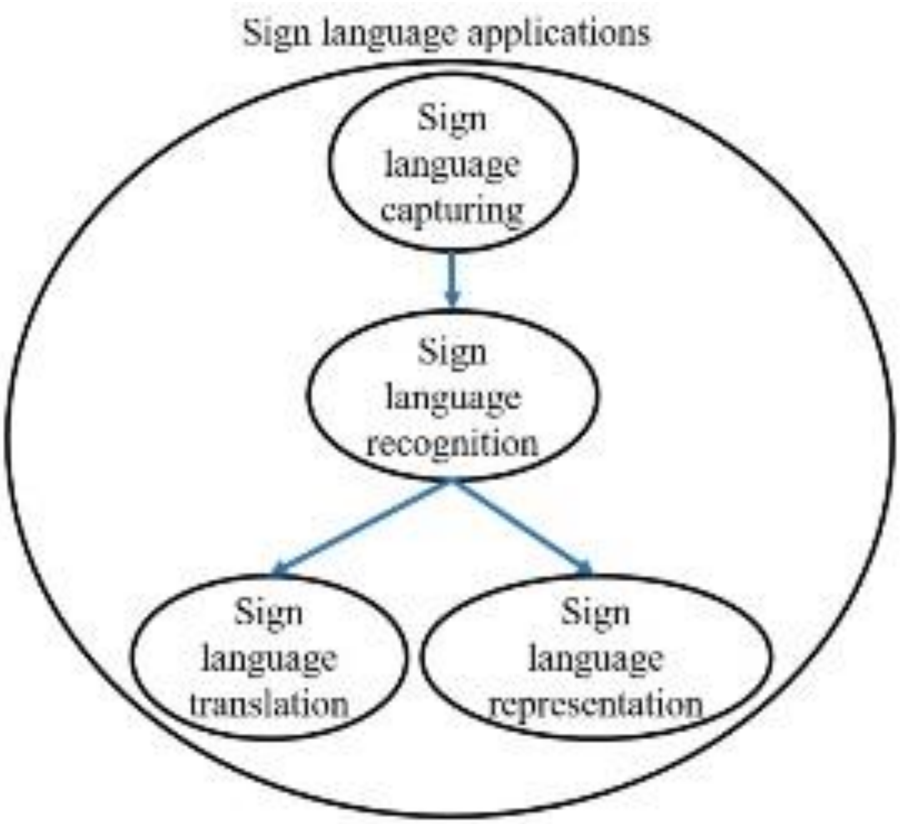
Company Overview:

IBM The International Business Machines Corporation (IBM), nicknamed Big Blue,[6] is an American multinational technology corporation headquartered in Armonk, New York and present in over 175 countries.[7][8] It specializes in computer hardware, middleware, and software, and provides hosting and consulting services in areas ranging from mainframe computers to nanotechnology. is a leading technology company that provides hardware, software, and services for enterprise applications. IBM has been a pioneer in the field of artificial intelligence and machine learning. The company has developed various AI and ML tools that help businesses automate their operations and make data-driven decisions.

Project Name- Sign Language recognition Model

About:

The project aims at building a machine learning model that will be able to classify the various hand gestures used for fingerspelling in sign language. In this user independent model, classification machine learning algorithms are trained using a set of image data and testing is done on a completely different set of data. For the image dataset, depth images are used, which gave better results than some of the previous literatures owing to the reduced pre-processing time. Various machine learning algorithms are applied on the datasets, including Convolutional Neural Network (CNN). An attempt is made to increase the accuracy of the CNN model by pre-training it on the ImageNet dataset.


Sign language applications
Sign language capturing
Sign language recognition
Sign language translation
Sign language representation

Tools Used for Building Sign Recognition Model

In the project, we used various tools and technologies for building the Sign Language Recognition system. Some of the tools that we used are:

OpenCV: A computer vision library used for image and video processing.

Mediapipe: A cross-platform framework for building multimodal applied machine learning pipelines.

Python: A programming language used for developing various applications.

Working methodology

1. <u>Importing necessary libraries:</u> The first step in the code is to import the necessary Python libraries, including pandas, numpy, and sklearn.

2. <u>Loading the data</u>: The next step is to load the dataset into the program using the pandas library. The read_csv() function is used to load the data from the CSV file.

3. <u>Data pre-processing:</u> The dataset is then preprocessed by removing any missing values or duplicates using the dropna() and drop_duplicates() functions respectively. Additionally, the LabelEncoder() function from the sklearn library is used to convert categorical features into numerical values.

4. <u>Splitting the data:</u> The data is then split into training and testing sets using the train_test_split() function from the sklearn library.

5. <u>Creating the model:</u> The next step involves creating a machine learning model using the RandomForestClassifier() function from the sklearn library.

6. <u>Training the model</u>: The model is then trained on the training data using the fit() function.

7. <u>Making predictions:</u> Once the model is trained, it is used to make predictions on the testing data using the predict() function.

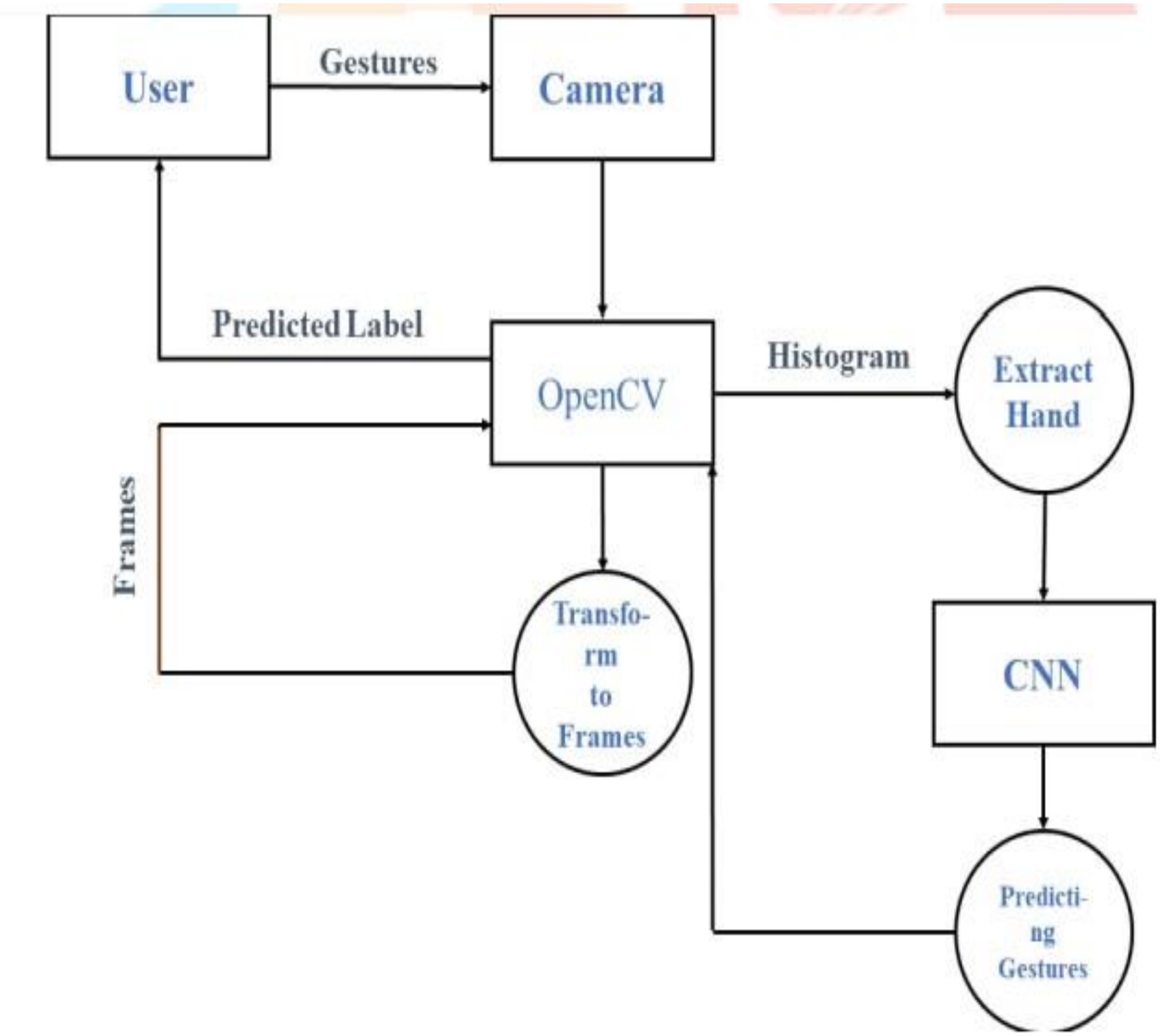
8. <u>Evaluating the model:</u> The performance of the model is evaluated using the accuracy_score() function from the sklearn library.

9. <u>Displaying the results:</u> Finally, the results of the model are displayed in the form of a confusion matrix and a classification report using the confusion_matrix() and classification_report() functions from the sklearn library, respectively.

WHO ARE THE END USERS?

The end users for an AI-Sign Language system would be people who are deaf or hard of hearing and use sign language as their primary means of communication. The system would be designed to help facilitate communication between these individuals and hearing individuals who may not understand sign language. Additionally, the system could be used by interpreters or educators who work with deaf or hard of hearing individuals to help them communicate more effectively.

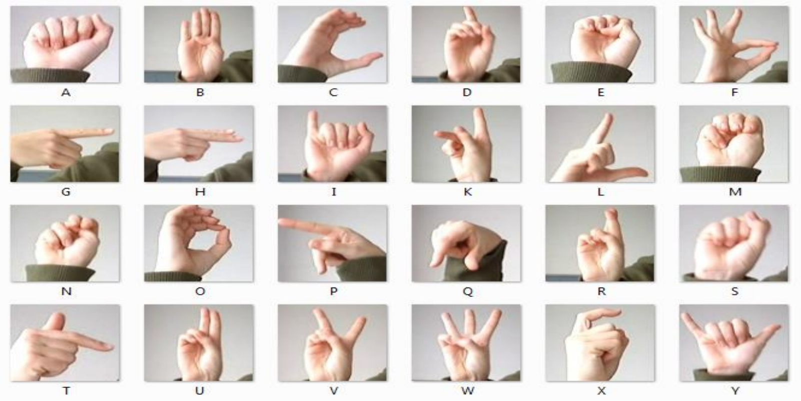
Flow diagram:



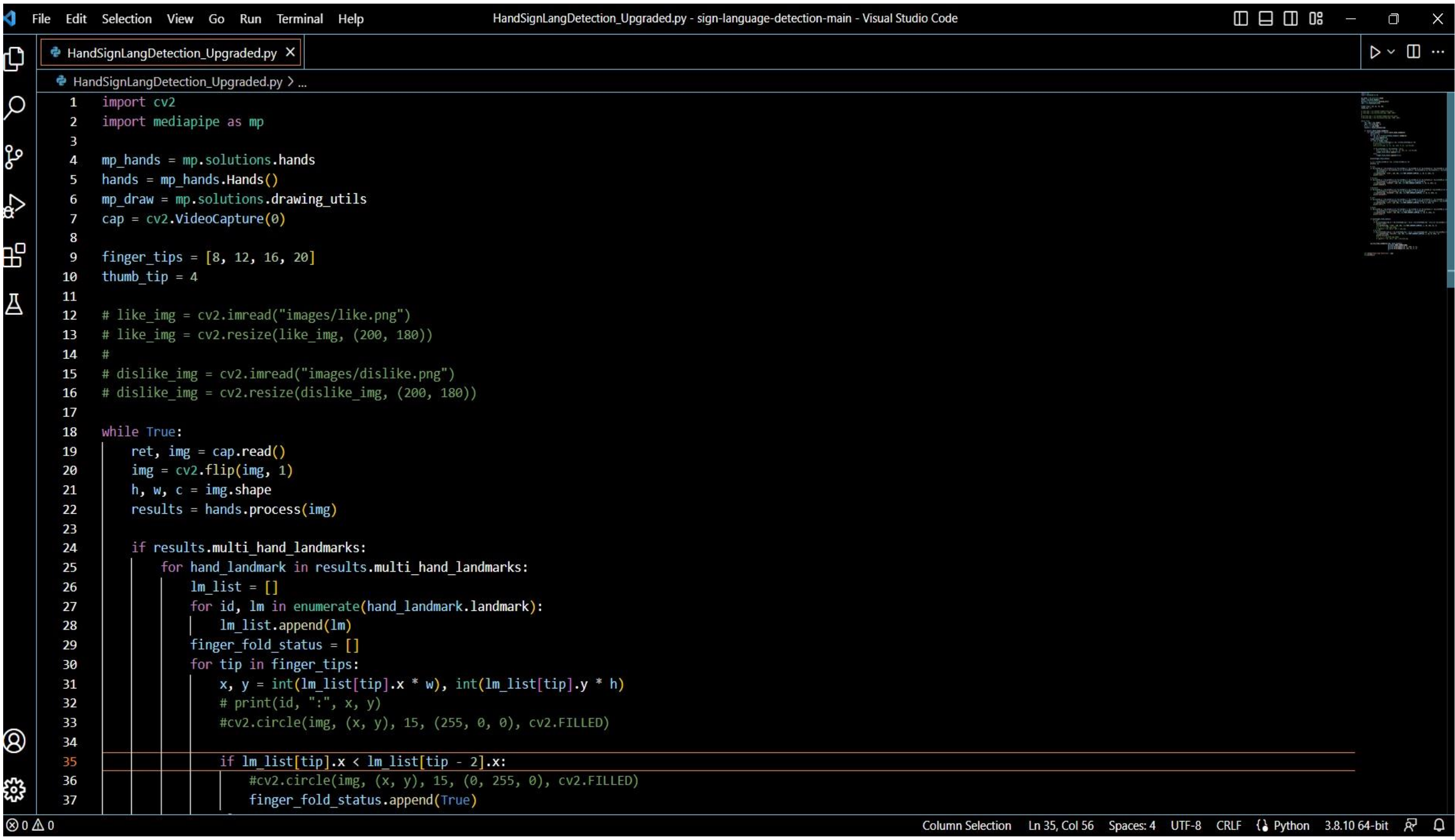
Problem Statement

The problem we were trying to solve in this project was the communication gap between people who are deaf or hard of hearing and those who can hear. Sign Language is one of the most commonly used means of communication among the deaf community. However, not everyone knows Sign Language, which makes communication difficult for them.

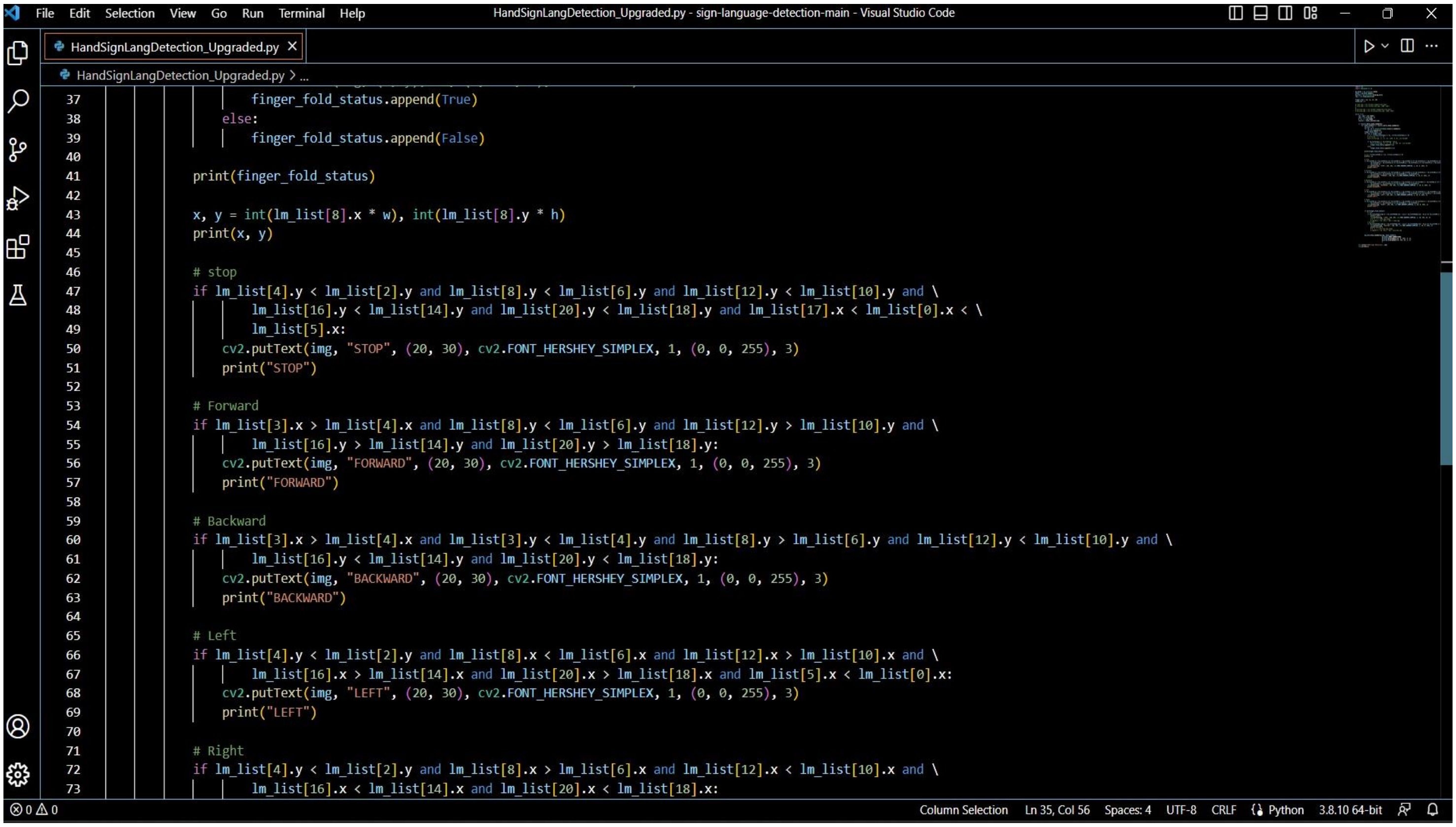
Solution



The solution we proposed to solve the communication gap was to develop a Sign Language Recognition system that can recognize different Sign Language Gestures and translate them into text. This system can be used by people who can hear to communicate with people who are deaf or hard of hearing. The system is also useful for people who are learning Sign Language as it can help them practice their gestures and improve their communication skills.

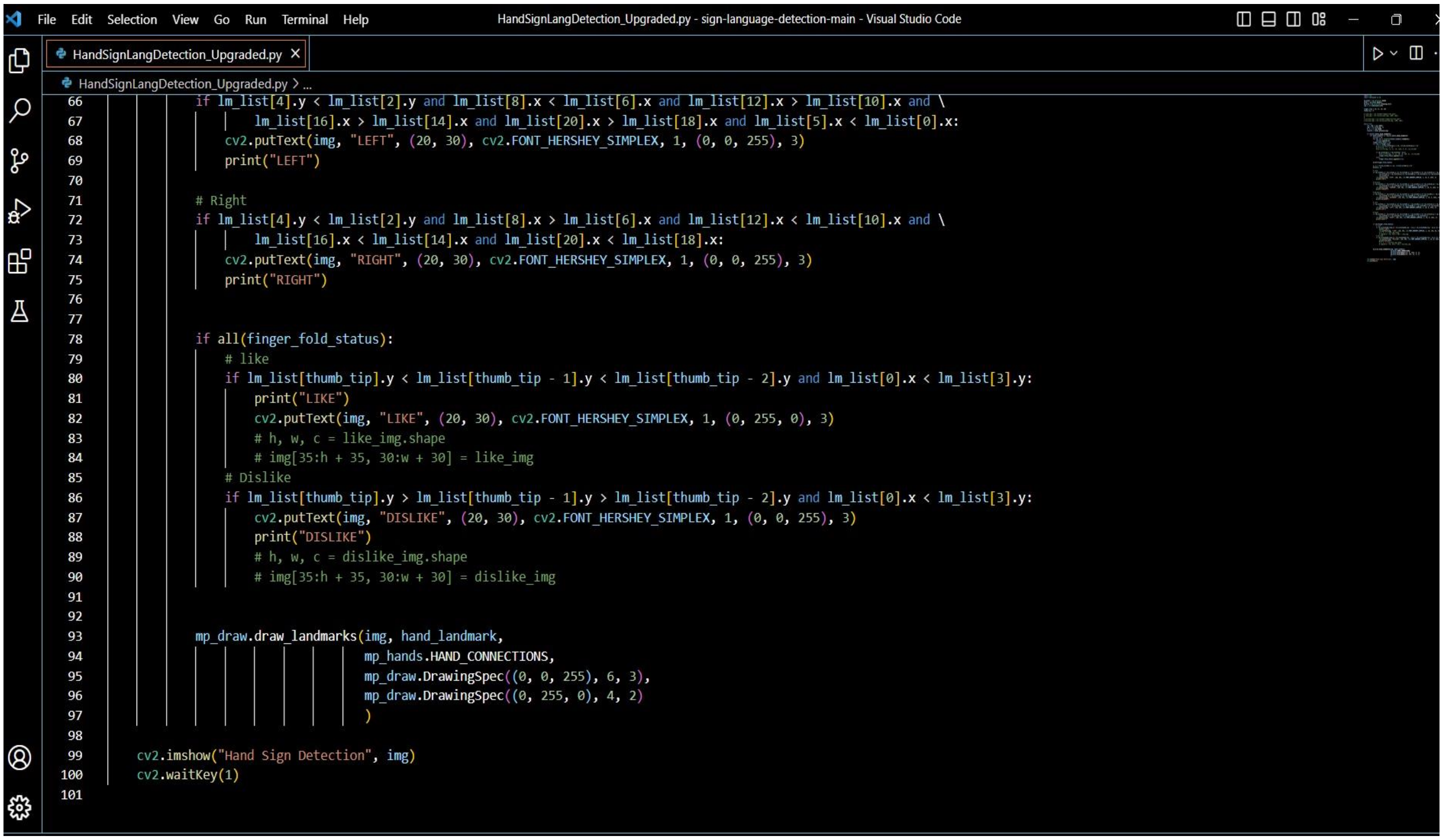

File
Edit
Selection
View
Go
Run
Terminal
Help
HandSignLangDetection_Upgraded.py - sign-language-detection-main - Visual Studio Code
HandSignLangDetection_Upgraded.py
HandSignLangDetection_Upgraded.py > ...
⊗ 0 ⚠ 0
Column Selection
Ln 35, Col 56
Spaces: 4
UTF-8
CRLF
{ } Python
3.8.10 64-bit

```python
import cv2
import mediapipe as mp

mp_hands = mp.solutions.hands
hands = mp_hands.Hands()
mp_draw = mp.solutions.drawing_utils
cap = cv2.VideoCapture(0)

finger_tips = [8, 12, 16, 20]
thumb_tip = 4

# like_img = cv2.imread("images/like.png")
# like_img = cv2.resize(like_img, (200, 180))
#
# dislike_img = cv2.imread("images/dislike.png")
# dislike_img = cv2.resize(dislike_img, (200, 180))

while True:
    ret, img = cap.read()
    img = cv2.flip(img, 1)
    h, w, c = img.shape
    results = hands.process(img)

    if results.multi_hand_landmarks:
        for hand_landmark in results.multi_hand_landmarks:
            lm_list = []
            for id, lm in enumerate(hand_landmark.landmark):
                lm_list.append(lm)
            finger_fold_status = []
            for tip in finger_tips:
                x, y = int(lm_list[tip].x * w), int(lm_list[tip].y * h)
                # print(id, ":", x, y)
                #cv2.circle(img, (x, y), 15, (255, 0, 0), cv2.FILLED)

                if lm_list[tip].x < lm_list[tip - 2].x:
                    #cv2.circle(img, (x, y), 15, (0, 255, 0), cv2.FILLED)
                    finger_fold_status.append(True)
```



File Edit Selection View Go Run Terminal Help
HandSignLangDetection_Upgraded.py - sign-language-detection-main - Visual Studio Code
HandSignLangDetection_Upgraded.py
HandSignLangDetection_Upgraded.py > ...
⊗ 0 ⚠ 0
Column Selection
Ln 35, Col 56
Spaces: 4
UTF-8
CRLF
{} Python
3.8.10 64-bit

```
                    finger_fold_status.append(True)
                else:
                    finger_fold_status.append(False)

            print(finger_fold_status)

            x, y = int(lm_list[8].x * w), int(lm_list[8].y * h)
            print(x, y)

            # stop
            if lm_list[4].y < lm_list[2].y and lm_list[8].y < lm_list[6].y and lm_list[12].y < lm_list[10].y and \
                    lm_list[16].y < lm_list[14].y and lm_list[20].y < lm_list[18].y and lm_list[17].x < lm_list[0].x < \
                    lm_list[5].x:
                cv2.putText(img, "STOP", (20, 30), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255), 3)
                print("STOP")

            # Forward
            if lm_list[3].x > lm_list[4].x and lm_list[8].y < lm_list[6].y and lm_list[12].y > lm_list[10].y and \
                    lm_list[16].y > lm_list[14].y and lm_list[20].y > lm_list[18].y:
                cv2.putText(img, "FORWARD", (20, 30), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255), 3)
                print("FORWARD")

            # Backward
            if lm_list[3].x > lm_list[4].x and lm_list[3].y < lm_list[4].y and lm_list[8].y > lm_list[6].y and lm_list[12].y < lm_list[10].y and \
                    lm_list[16].y < lm_list[14].y and lm_list[20].y < lm_list[18].y:
                cv2.putText(img, "BACKWARD", (20, 30), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255), 3)
                print("BACKWARD")

            # Left
            if lm_list[4].y < lm_list[2].y and lm_list[8].x < lm_list[6].x and lm_list[12].x > lm_list[10].x and \
                    lm_list[16].x > lm_list[14].x and lm_list[20].x > lm_list[18].x and lm_list[5].x < lm_list[0].x:
                cv2.putText(img, "LEFT", (20, 30), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255), 3)
                print("LEFT")

            # Right
            if lm_list[4].y < lm_list[2].y and lm_list[8].x > lm_list[6].x and lm_list[12].x < lm_list[10].x and \
                    lm_list[16].x < lm_list[14].x and lm_list[20].x < lm_list[18].x:
```



File Edit Selection View Go Run Terminal Help
HandSignLangDetection_Upgraded.py - sign-language-detection-main - Visual Studio Code
HandSignLangDetection_Upgraded.py
HandSignLangDetection_Upgraded.py > ...

```
            if lm_list[4].y < lm_list[2].y and lm_list[8].x < lm_list[6].x and lm_list[12].x > lm_list[10].x and \
                    lm_list[16].x > lm_list[14].x and lm_list[20].x > lm_list[18].x and lm_list[5].x < lm_list[0].x:
                cv2.putText(img, "LEFT", (20, 30), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255), 3)
                print("LEFT")

            # Right
            if lm_list[4].y < lm_list[2].y and lm_list[8].x > lm_list[6].x and lm_list[12].x < lm_list[10].x and \
                    lm_list[16].x < lm_list[14].x and lm_list[20].x < lm_list[18].x:
                cv2.putText(img, "RIGHT", (20, 30), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255), 3)
                print("RIGHT")


            if all(finger_fold_status):
                # like
                if lm_list[thumb_tip].y < lm_list[thumb_tip - 1].y < lm_list[thumb_tip - 2].y and lm_list[0].x < lm_list[3].y:
                    print("LIKE")
                    cv2.putText(img, "LIKE", (20, 30), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 3)
                    # h, w, c = like_img.shape
                    # img[35:h + 35, 30:w + 30] = like_img
                # Dislike
                if lm_list[thumb_tip].y > lm_list[thumb_tip - 1].y > lm_list[thumb_tip - 2].y and lm_list[0].x < lm_list[3].y:
                    cv2.putText(img, "DISLIKE", (20, 30), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255), 3)
                    print("DISLIKE")
                    # h, w, c = dislike_img.shape
                    # img[35:h + 35, 30:w + 30] = dislike_img


            mp_draw.draw_landmarks(img, hand_landmark,
                                   mp_hands.HAND_CONNECTIONS,
                                   mp_draw.DrawingSpec((0, 0, 255), 6, 3),
                                   mp_draw.DrawingSpec((0, 255, 0), 4, 2)
                                   )

    cv2.imshow("Hand Sign Detection", img)
    cv2.waitKey(1)
```

There are several uses and advantages of AI-Sign Language systems, including:

1. Improved Communication: AI-Sign Language systems can help bridge the communication gap between people who are deaf or hard of hearing and those who are hearing, by facilitating real-time translation of sign language into written or spoken language, and vice versa.

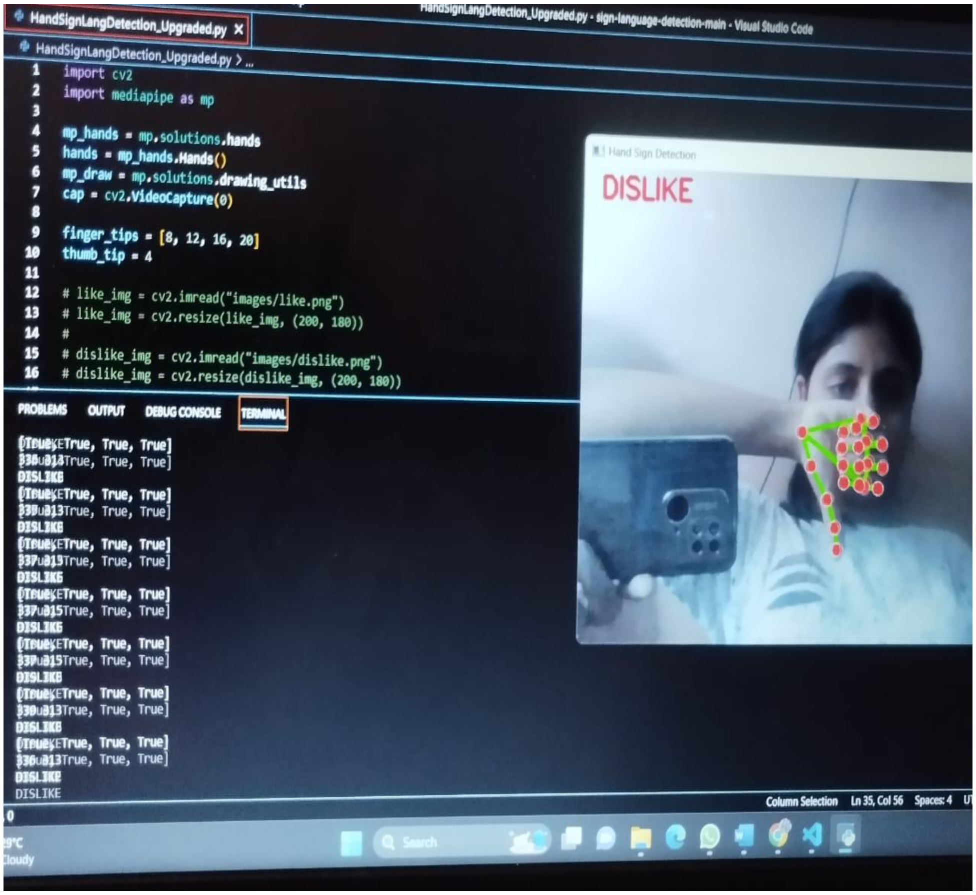
2. Accessibility: AI-Sign Language systems make it easier for people who are deaf or hard of hearing to access a range of services, including healthcare, education, and employment, by removing communication barriers.

3. Time Efficiency: AI-Sign Language systems can help save time for both the sign language user and the hearing individual, by providing instant translation without the need for an interpreter or written notes.

4. Consistency: AI-Sign Language systems can provide consistent and accurate translations of sign language, which can help prevent misunderstandings and errors that may occur with human interpretation.

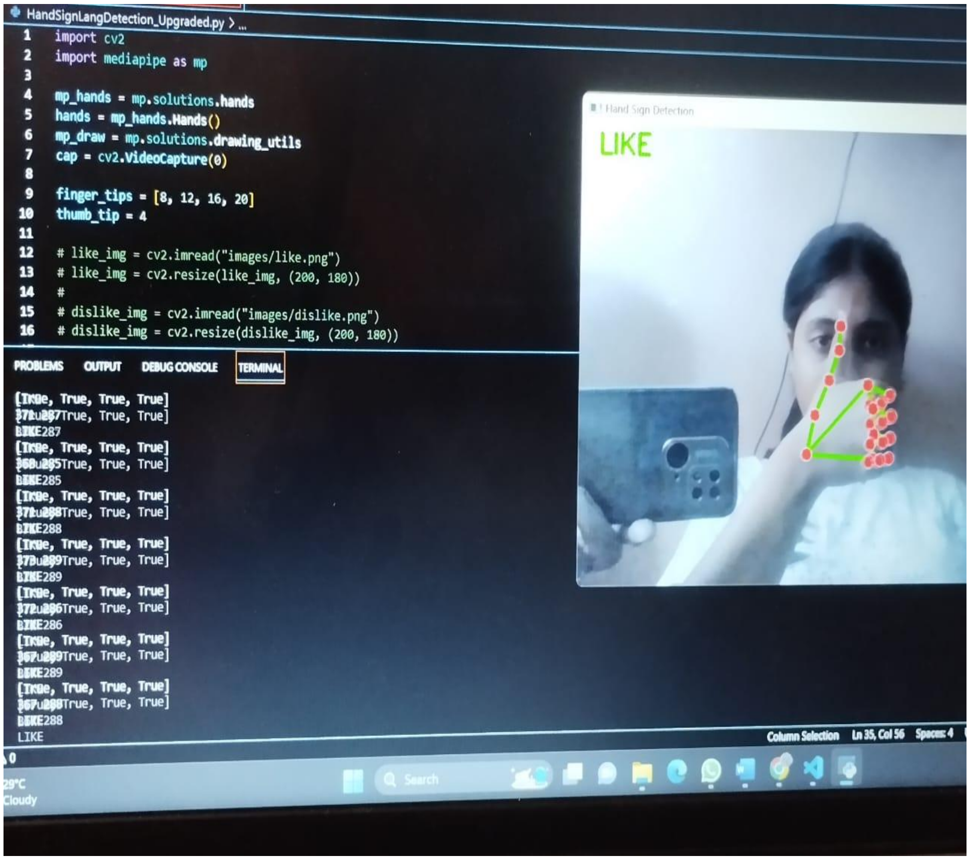
5. Cost Efficiency: AI-Sign Language systems can be cost-effective compared to hiring a human interpreter for every interaction, especially in situations where an interpreter is not readily available.

OUTPUT:


HandSignLangDetection_Upgraded.py ×
HandSignLangDetection_Upgraded.py - sign-language-detection-main - Visual Studio Code
HandSignLangDetection_Upgraded.py > ...
Hand Sign Detection
DISLIKE
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
Column Selection
Ln 35, Col 56
Spaces: 4
29°C
Cloudy
Search

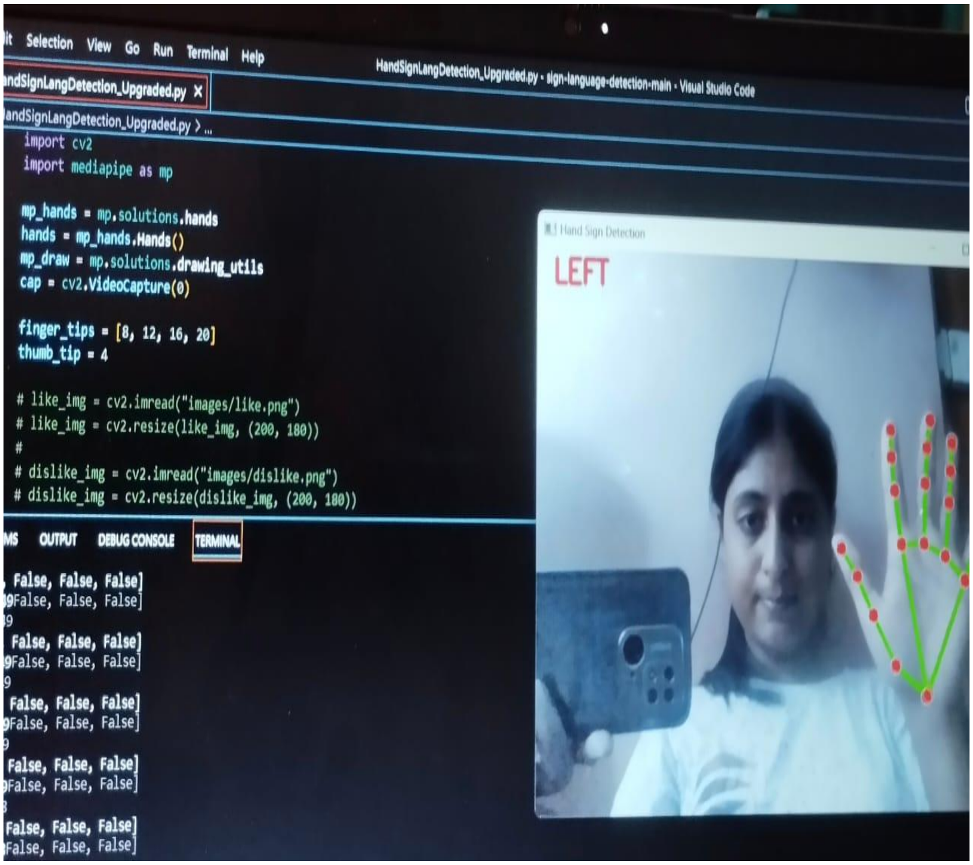
```
import cv2
import mediapipe as mp

mp_hands = mp.solutions.hands
hands = mp_hands.Hands()
mp_draw = mp.solutions.drawing_utils
cap = cv2.VideoCapture(0)

finger_tips = [8, 12, 16, 20]
thumb_tip = 4

# like_img = cv2.imread("images/like.png")
# like_img = cv2.resize(like_img, (200, 180))
#
# dislike_img = cv2.imread("images/dislike.png")
# dislike_img = cv2.resize(dislike_img, (200, 180))
```

```
[TrueKETrue, True, True]
338u013True, True, True]
DISLIKE
[TrueKETrue, True, True]
339u013True, True, True]
DISLIKE
[TrueKETrue, True, True]
337u015True, True, True]
DISLIKE
[TrueKETrue, True, True]
337u015True, True, True]
DISLIKE
[TrueKETrue, True, True]
337u015True, True, True]
DISLIKE
[TrueKETrue, True, True]
330u013True, True, True]
DISLIKE
[TrueKETrue, True, True]
336u013True, True, True]
DISLIKE
DISLIKE
```



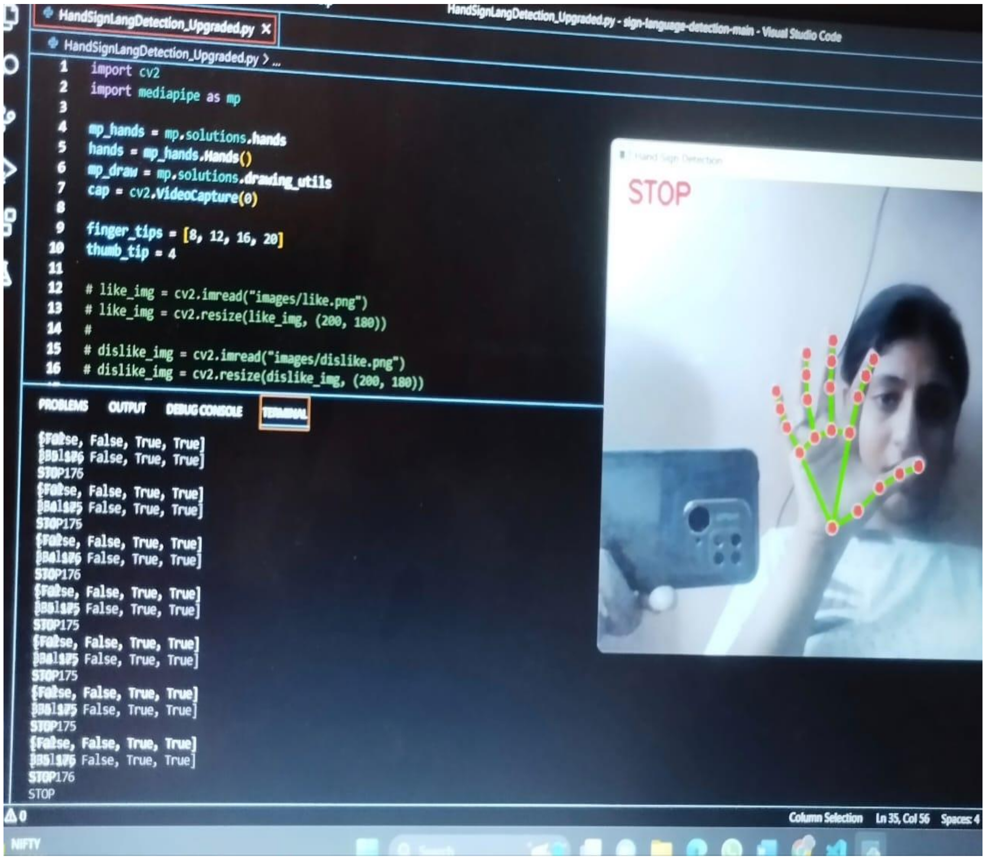
```
import cv2
import mediapipe as mp

mp_hands = mp.solutions.hands
hands = mp_hands.Hands()
mp_draw = mp.solutions.drawing_utils
cap = cv2.VideoCapture(0)

finger_tips = [8, 12, 16, 20]
thumb_tip = 4

# like_img = cv2.imread("images/like.png")
# like_img = cv2.resize(like_img, (200, 180))
#
# dislike_img = cv2.imread("images/dislike.png")
# dislike_img = cv2.resize(dislike_img, (200, 180))
```

```
[True, True, True, True]
371u287True, True, True]
LIKE287
[True, True, True, True]
368u285True, True, True]
LIKE285
[True, True, True, True]
371u288True, True, True]
LIKE288
[True, True, True, True]
373u289True, True, True]
LIKE289
[True, True, True, True]
372u286True, True, True]
LIKE286
[True, True, True, True]
367u289True, True, True]
LIKE289
[True, True, True, True]
367u288True, True, True]
LIKE288
LIKE
```


it Selection View Go Run Terminal Help
HandSignLangDetection_Upgraded.py - sign-language-detection-main - Visual Studio Code
andSignLangDetection_Upgraded.py ×
andSignLangDetection_Upgraded.py > ...
Hand Sign Detection
LEFT
MS OUTPUT DEBUG CONSOLE TERMINAL

```
import cv2
import mediapipe as mp

mp_hands = mp.solutions.hands
hands = mp_hands.Hands()
mp_draw = mp.solutions.drawing_utils
cap = cv2.VideoCapture(0)

finger_tips = [8, 12, 16, 20]
thumb_tip = 4

# like_img = cv2.imread("images/like.png")
# like_img = cv2.resize(like_img, (200, 180))
#
# dislike_img = cv2.imread("images/dislike.png")
# dislike_img = cv2.resize(dislike_img, (200, 180))
```

```
, False, False, False]
9False, False, False]
9
 False, False, False]
9False, False, False]
9
 False, False, False]
9False, False, False]
9
 False, False, False]
9False, False, False]
3
 False, False, False]
False, False, False]
```


HandSignLangDetection_Upgraded.py ×
HandSignLangDetection_Upgraded.py - sign-language-detection-main - Visual Studio Code
HandSignLangDetection_Upgraded.py > ...
Hand Sign Detection
STOP
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Column Selection Ln 35, Col 56 Spaces: 4
NIFTY

```
import cv2
import mediapipe as mp

mp_hands = mp.solutions.hands
hands = mp_hands.Hands()
mp_draw = mp.solutions.drawing_utils
cap = cv2.VideoCapture(0)

finger_tips = [8, 12, 16, 20]
thumb_tip = 4

# like_img = cv2.imread("images/like.png")
# like_img = cv2.resize(like_img, (200, 180))
#
# dislike_img = cv2.imread("images/dislike.png")
# dislike_img = cv2.resize(dislike_img, (200, 180))
```

```
[False, False, True, True]
[Fa176 False, True, True]
STOP176
[False, False, True, True]
[Fa175 False, True, True]
STOP175
[False, False, True, True]
[Fa176 False, True, True]
STOP176
[False, False, True, True]
[Fa175 False, True, True]
STOP175
[False, False, True, True]
[Fa175 False, True, True]
STOP175
[False, False, True, True]
[Fa175 False, True, True]
STOP175
[False, False, True, True]
[Fa176 False, True, True]
STOP176
STOP
```

From: Gayatri pramod kumar Samal
Sent: 19 January 2023 09:52
To: Sami Ahmad
Subject: Re: AICTE 8 Weeks Internship Batch 4_20.03.2023

I accept the offer.

On Thu, 19 Jan, 2023, 9:28 am Sami Ahmad, <sami@edunetfoundation.org> wrote:

Internship Offer Letter

Dear Mr./Ms. **GAYATRI PRAMOD KUMAR SAMAL**

Student ID:STU62a57bab574bb1655012267 Internship ID:INTERNSHIP_16699559116389814793bca Date: 20.01.2023

We would like to congratulate you on being selected for an Internship under IBM SkillsBuild Program on **Artificial Intelligence.** As an intern, you will learn and demonstrate skills that will enhance your employability and your confidence in the subject area.

IBM SkillsBuild is an e-learning platform that helps job seekers gain the foundational skills necessary to prepare for entry-level positions while earning credentials and receiving guidance for industry experts at no cost. The IBM SkillsBuild internship is brought to you by the Edunet Foundation. We are pleased to offer you an internship for a period of 8 weeks beginning from **20.01.2023 to 17.03.2023**.

During this internship, you will work in a team and be assigned a mentor who will guide you to identify a solution to the problem and develop it into a project. The internship will be providing the following benefits:

- Access to IBM SkillsBuild e-Learning Platform with over 80,000 curated courses on technical and professional skills
- Opportunity to accelerate your learning in project-based, collaborative environments
- Learners can experience masterclasses led by Subject Matter Experts
- Showcase your skills by solving real-world challenges

- Opportunity to accelerate your learning in project-based, collaborative environments
- Learners can experience masterclasses led by Subject Matter Experts
- Showcase your skills by solving real-world challenges

- Certification from IBM and Edunet to enhance your value to future employers

Timeline and project:

The structure of the IBM SkillsBuild internship camp is as follows

Week	Outcome
Week 1-2	Learning the domain skills & Problem Identification
Week 3-4	Solution Design + SkillsBuild Courses
Week 5-6	Solution Development + SkillsBuild Courses
Week 7-8	Final Project/ Website & Small Presentation

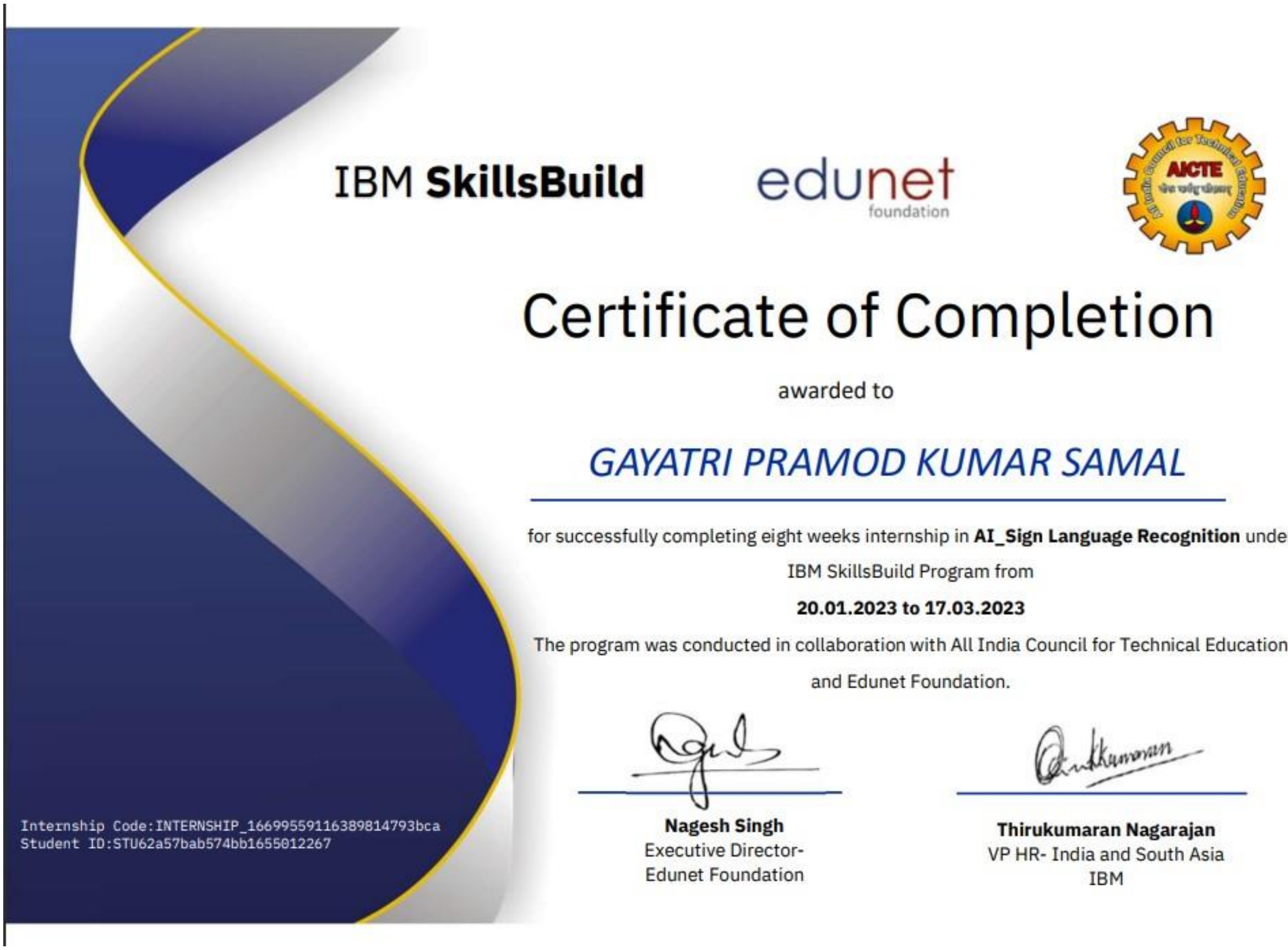
Criteria for certification:

- Participation in weekly sessions with mentors is mandatory
- Completion of tasks/milestones on a weekly basis
- Completion of SkillsBuild courses
- Project presentation

Stipend:

There will be NO stipend for this internship. If you agree to the above terms of the offer, please indicate acceptance of the offer letter and below undertaking to the undersigned.

☒ Loading...



IBM SkillsBuild

edunet
foundation

Certificate of Completion

awarded to

GAYATRI PRAMOD KUMAR SAMAL

for successfully completing eight weeks internship in **AI_Sign Language Recognition** under IBM SkillsBuild Program from

20.01.2023 to 17.03.2023

The program was conducted in collaboration with All India Council for Technical Education and Edunet Foundation.

Nagesh Singh
Executive Director-
Edunet Foundation

Thirukumaran Nagarajan
VP HR- India and South Asia
IBM

Internship Code:INTERNSHIP_16699559116389814793bca
Student ID:STU62a57bab574bb1655012267

Conclusion:

The Sign Language Recognition system project was an enriching experience that helped me learn various machine learning and artificial intelligence concepts in a real-world setting. I got to work with some of the most advanced AI and ML tools and technologies, which helped me develop my skills and gain practical knowledge. The project also highlighted the potential of AI and ML in solving real-world problems and making a positive impact on society.

Perhaps most importantly, this internship or project taught me the value of perseverance and dedication. Working in the field of AI and ML requires a tremendous amount of patience, as well as an unwavering commitment to understanding complex data sets and algorithms. But the rewards of such work are immeasurable, as the insights and solutions that can be gleaned from this type of work have the potential to transform industries and improve people's lives in countless ways.

As I look back on my time with this internship or project, I am grateful for the opportunities it afforded me, and I am excited about the future of AI and ML. I urge all of you to explore the possibilities of this exciting field, and to seek out opportunities to gain hands-on experience and to work alongside experts in the field. With dedication, hard work, and a commitment to innovation, the possibilities are truly endless.

Thank you.