SUMMER PROJECT REPORT ON

SIGN LANGUAGE TRANSLATOR

BY

Abdul Azim Rajmohamed (501801)
Abhishek Manilal Gupta (501824)
Ansh Mitesh Chhadva (501816)
Gladina Raymond (501821)

DEPARTMENT OF INFORMATION TECHNOLOGY

FR. CONCEICAO INSTITUTE OF TECHNOLOGY

Sector 9A, Vashi, Navi Mumbai-400703

University of Mumbai

2020-2021

ABSTRACT

According to the World Health Organization (WHO), 466 million people across the world have disabling hearing loss (over 5% of the world's population), of whom 34 million are children. There are only about 250 certified sign language interpreters in India for a deaf population of around 7 million.

With these significant statistics, the need for developing a tool for smooth flow of communication between abled and people with speech/hearing impairment is very high.

Our application promises to secure a two way conversation, as it deploys machine learning and deep learning models to convert sign language to speech/text. The opposite receiver can either speak or text his response, which will then be visible to the disabled person in the form of text. The client can make use of the tutorials and learn the basic functioning of the application and ASL.

This system eliminates the need of an interpreter and the traditional methods of pen and paper can also be discarded.

This application ensures the automation of communication and thereby provides a solution to the hurdles faced by hearing/speech impaired people.

TABLE OF CONTENT

TABLE FOR FIGURES

CHAPTER 1

INTRODUCTION

❑ BACKGROUND

With the increase of innovations and technology, life has become significantly easy for humans. The sudden surge of growth in tech has left many overjoyed and overwhelmed because of the good fruits it bears. It has paved way for the poor to become rich, the sick to become strong, the disabled to experience the life of an abled. People with speech/hearing impairment have always found it difficult to communicate and mingle but with technology, that barrier has also been destroyed. They can now communicate without any difficulty and find themselves in a public setting, communicating confidently.

❑ MOTIVATION/NEED/PURPOSE

To assist differently abled people with respect to speaking and hearing, translators were always a must. Since the sign language is not known and understood by many, a translator was always necessary. However, it's not very convenient for a translator to be available at all times, and sometimes not convenient in particular settings. To overcome this, the ***Sign Language Translator*** is a tool to eliminate the need of a translator. The application translates sign language and enables the opposite person to understand what is to be conveyed. It also helps vice versa i.e. what is spoken will be relayed in text as well.

❑ PROBLEM DEFINITION

The traditional methods of communicating with the deaf and mute are really not convenient in many aspects. The alternatives that are available to break this barrier have definite flaws.

An interpreter is not always available and this method is not cost efficient either. The pen and paper method is highly unprofessional and also time consuming.

Texting and messaging are fine to a certain extent but still does not tackle the bigger problem at hand. This has created a grave need to develop a solution to destroy the barricade of communication effectively.

❑ SCOPE

Sign Language Translator enables the hearing impaired user to communicate efficiently in sign language, and the application will translate the same into text/speech. The user has to train the model, by recording the sign language gestures and then label the gesture. The user can then use the saved and recorded gestures while speaking to other people.

❑ PROPOSED SYSTEM FEATURES

To break the above mentioned barricade, the application promises to secure a perfect communication. The hearing impaired user will be given access to full-fledged tutorials to guide him/her to use the application on the website. The tutorials page for sign language consists of 70+ most common used phrases like "Open the door " , all English alphabet in sign language and some signs for objects and place names like Mumbai

TUTORIALS

Tutorials are provided in assisting the user for a smooth learning process it helps to learn ASL alphabets , Most commonly used phrases in day to day communication and some signs of popular places like 'Mumbai' and objects like 'Police Station' , etc

TRAINING YOUR OWN SIGN LANGUAGE

The user can then begin to record gestures and train the model. The user will begin to do the gestures in Sign Language and the camera will record the same. The user then has to assign a label to that particular gesture.

As an additional feature, the user can record an entire statement and assign a label to that, which he thinks is used very often. This helps in user ease and efficiency

Once a set of gestures are recorded, the user can then communicate with other people using the recorded set of gestures. He/she will show the gestures of the message that has to be conveyed, and the trained model will convey the message in the form of text or speech to the opposite user with 100% accuracy.

The opposite user can respond via text or speech which will then be relayed back.

The user can also save the model and then upload later, getting the same prediction without training.

The entire project runs on the browser and being lightweight it is completely client side without any involvement of the server to slow down the process .

Fig 1 : Working of Sign Translator

PRACTICE YOUR SKILLS

When the user communicates with an opposite party using sign language, the pre-trained model then uses the recorded gestures to relay the desired message in the form of text or speech to the receiver.

The Model currently supports 21 alphabets/words signs in ASL , It can be used by beginner users to practice his skills against pre-trained models.

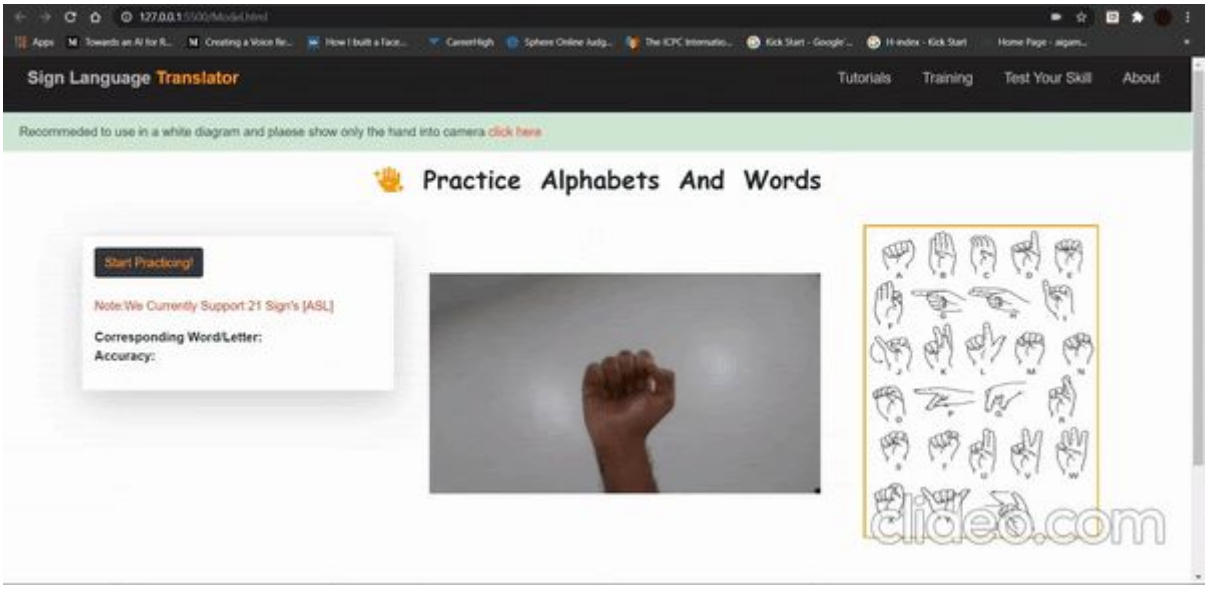

Fig 2 : Working of Practice Page

❑ OBJECTIVES

1) Eliminate the need of an interpreter.

2) Ease the communication flow for hearing/speech impaired people through our model predictions and text to speech system.

3) Provide high quality video tutorials for most commonly used phrases and English sign alphabets (ASL).

4) Provide a portal for practicing the learned alphabets.

5) Ability to create new signs for any text or sentence in the browser (client side).

❑ ISSUES/LIMITATIONS

1) The user has to himself provide the dataset by making the sign language gestures and then label them. When the user records a large set of gestures.

2) For practicing skills it is necessary for the user to have a white background with no other object in the frame of the camera apart from the hand

.

CHAPTER 2

LITERATURE SURVEY

❑ EXISTING SYSTEMS

1. The first sign-language glove to gain any notoriety came out in 2001. A high-school student from Colorado, Ryan Patterson, fitted a leather golf glove with 10 sensors that monitored finger position, then relayed finger spellings to a computer which rendered them as text on a screen. In 2002, the public-affairs office of the National Institute on Deafness and Other Communicative Disorders effused said about Patterson, The glove doesn't translate anything beyond individual letters, certainly not the full range of signs used in American Sign Language, and works only with the American Manual Alphabet.

   https://www.theatlantic.com/technology/archive/2017/11/

2. MotionSavvy is building a tablet which detects when a person is using ASL and converts it to text or voice. The software also has voice recognition through the tablet's mic, which allows a hearing person to respond with voice to the person signing. It then converts their voice into text, which the hearing-impaired receiver can understand.

   https://techcrunch.com/2014/06/06/motionsavvy-is-a-tablet-app-that-understands-sign-language/

3. The application, Lingo jam only translates alphabetically. The manual sign language or fingerspelling is followed and not the universal sign language . Each letter is translated as it is, and displayed over text only.

   https://lingojam.com/AmericanSignLanguageTranslator

4. A Netherlands-based start-up has developed an artificial intelligence (AI) powered smartphone app for deaf and mute people, which it says offers a low-cost and superior approach to translating sign language into text and speech in real time.

   The easy-to-use innovative digital interpreter dubbed as "Google translator for the deaf and mute" works by placing a smartphone in front of the user while the app translates gestures or sign language into text and speech.

   https://economictimes.indiatimes.com/

CHAPTER 3

MODEL DESIGN

Sign Training Model

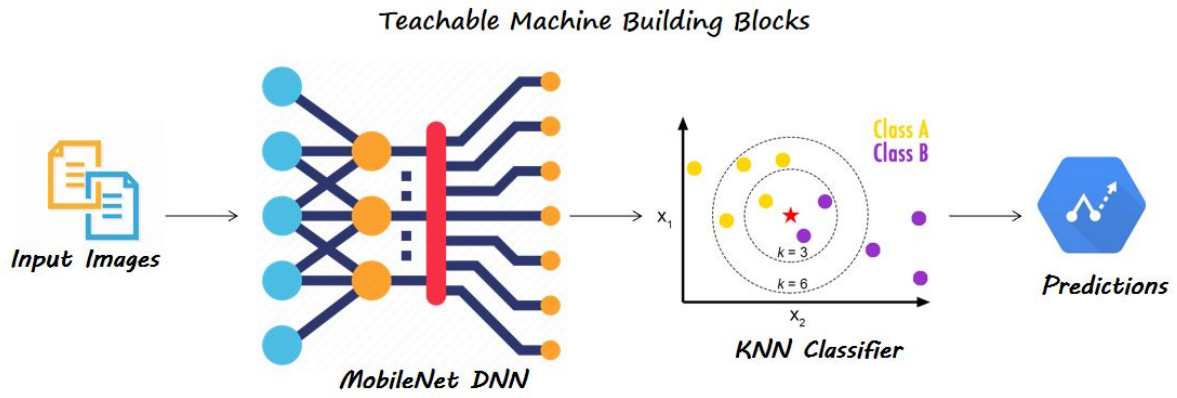

Fig 3 : Sign Training Model

Sign Training Model lets anyone build their own image classification model with no coding required. All you need is a webcam.

The approach we're going to take is called transfer learning. This technique starts with an already trained model and specializes it for the task at hand. This lets you train far more quickly and with less data than if you were to train from scratch.

We bootstrap our model from a pre-trained model called MobileNet. Our system will learn to make predictions using our own classes that were never seen by MobileNet. We do this by using the activations produced by this pretrained model, which informally represent high-level semantic features of the image that the model has learned.

The pretraining is so effective that we don't have to do anything fancy like train another neural network, but instead we just use a nearest neighbors approach. What we do is feed an image through MobileNet and find other examples in the dataset that have similar activations to this image. In practice, this is noisy, so instead we choose the k-nearest neighbors and choose the class with the most representation.
By bootstrapping our model with MobileNet and using k-nearest neighbors, we can train a realistic classifier in a short amount of time, with very little data, all in the browser. Doing this fully end-to-end, from pixels to prediction, won’t require too much time and data for an interactive application.

Alphabets and Phrases Model

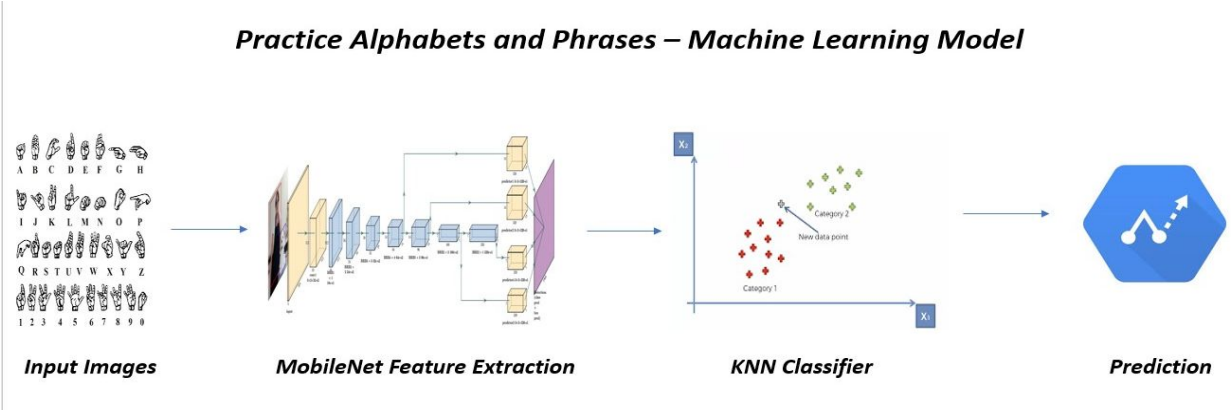

Fig 4: Practice Alphabet and Phrases model

The Practice Sign Model lets you practice your sign language skills which you learned on our tutorial page.

The above model is mainly divided into 3 parts:

1) **Dataset**
2) **MobileNet Feature Extraction**
3) **Knn Classifier**

Starting with the dataset it consists of above 1000 input images of different sign languages belonging to 21 different classes. Each image is of size 64 x 64.

The approach we're going to take is called transfer learning. This technique starts with an already trained model and specializes it for the task at hand. This lets you train far more quickly and with less data than if you were to train from scratch.

We bootstrap our model from a pre-trained model called MobileNet. Our system will learn to make predictions using our own classes that were never seen by MobileNet. We do this by using the feature extraction technique.

Feature extraction is a process of dimensionality reduction by which an initial set of raw data is reduced to more manageable groups for processing. A characteristic of these large data sets is a large number of variables that require a lot of computing resources to process.t.

The pretraining is so effective that we don't have to do anything fancy like train another neural network, but instead we just use a nearest neighbors approach.

CHAPTER 4

SYSTEM REQUIREMENT

SYSTEM REQUIREMENTS

- Web Browser with Internet Connection.
- Web Camera

CHAPTER 5

IMPLEMENTATION DETAILS

Visit Our Website : [Click here]

Learn New Sign Language Alphabets (ASL) and phrases

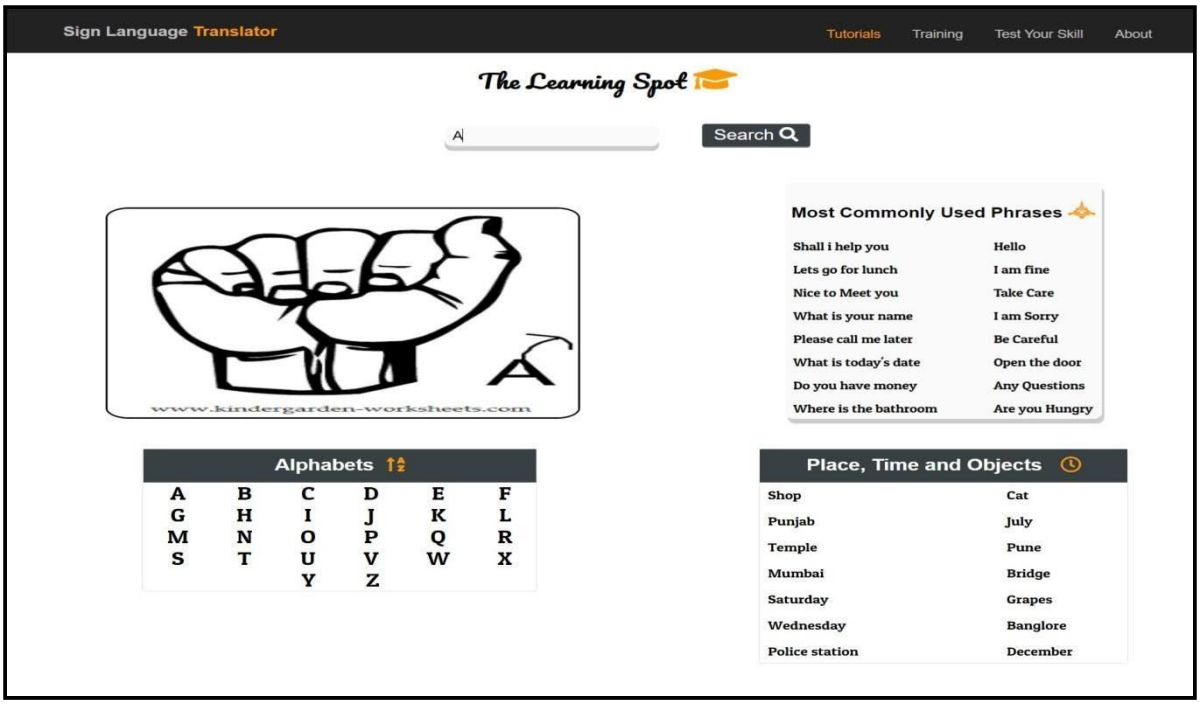

Fig 5: Website Tutorial Page

Select any Phrases or alphabets from " Most Commonly Used Phrases " section , alphabets from " Alphabets " section or any phrase from " Place , Time and Objects " section and learn through high quality videos and images.

Make Your Own Sign Language

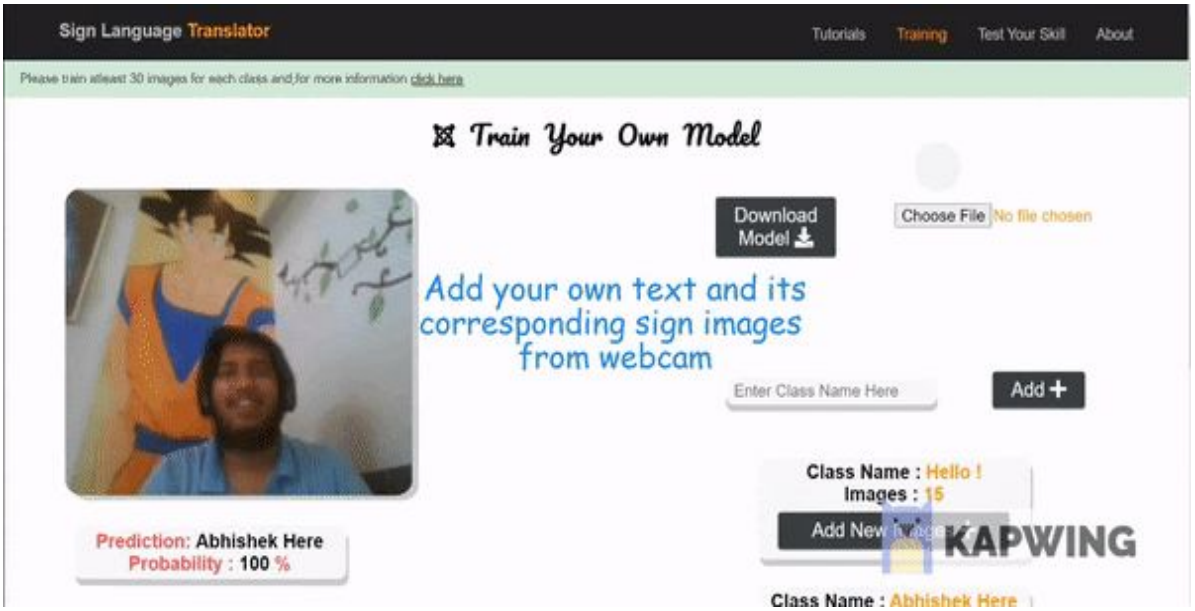

Fig 6: Website – Train Your Model Page

Write your text that is your label for that particular class and click on the “Add ” button, after that using a webcam as input click on the “ Add New Images ” button and add 10-40 images per class. You will be able to observe the real-time predictions below the video

Click on “Speak” button to convert the predicted text into speech using your favourite voice

Practice Alphabets and Phrases

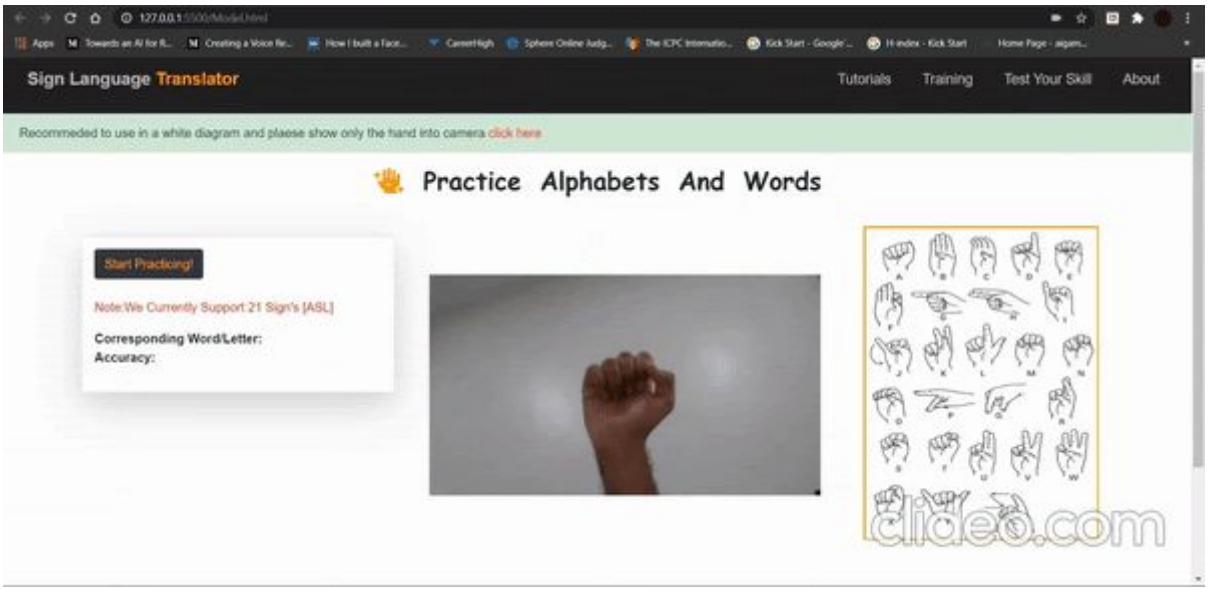

Fig 7: Website – Practice Alphabets and Words Page

Click on the “ Start Practicing” button and place your hand in front of the camera against white background and start practicing your signs (Alphabets) . You will be able to observe the prediction below the button.

CHAPTER 7

CONCLUSION
&
FUTURE SCOPE

❑ CONCLUSION

This project was undertaken to solve the underlying issue faced by hearing and speech impaired people. They often don't even stand a chance in the competitive global arena because of communication hurdles.

This project, however, helps in eradicating the social stigma of them not able to participate in many domains and successfully gives them confidence to stand upright in any field they want.

The application provides the necessary platform to communicate with much ease and gives them the ability to interact without any external help.

The need of an interpreter is eradicated and the smooth flowing of a conversation is well developed.

❑ FUTURE SCOPE

The model and text to speech can be embedded into a video calling system. Thereby allowing the user to show the gestures and the receiver on the call will receive the message in the form of text or speech. While the receiver responds, the message will be relayed to the hearing/speech impaired user via text (subtitles).

CHAPTER 8

APPENDIX: CODE SAMPLES

Code Samples

```
    const createKNNClassifier = async () => {
        console.log('Loading KNN Classifier');
        return await knnClassifier.create();
    };
    const createMobileNetModel = async () => {
        console.log('Loading Mobilenet Model');
        return await mobilenet.load();
    };
    const createWebcamInput = async () => {
        console.log('Loading Webcam Input');
        const webcamElement = await document.getElementById('webcam');
        return await tf.data.webcam(webcamElement);
    };

    const mobilenetModel = await createMobileNetModel();
    const knnClassifierModel = await createKNNClassifier();
    const webcamInput = await createWebcamInput();
    var preloader = document.getElementById("loading");
```

Fig 8: Code Example

```
  // Extract the already learned features from MobileNet
  featureExtractor = ml5.featureExtractor('MobileNet', modelReady);
  const knnClassifier = ml5.KNNClassifier();


  // Create a new classifier using those features and give the video we want to use
  const options = { numLabels: 21 };
  classifier = featureExtractor.classification(video, options);
  // Set up the UI buttons
  setupButtons();
}

// A function to be called when the model has been loaded
async  function modelReady() {
  select('#modelStatus').html('MobileNet Loaded!');
  // If you want to load a pre-trained model at the start

  await classifier.load('model.json', function() {
      select('#modelStatus').html('Custom Model Loaded!');
    });
}

// Classify the current frame.
function classify() {
  classifier.classify(gotResults);
}
```

Fig 9: Code Example

```
const imageClassificationWithTransferLearningOnWebcam = async () => {
    console.log("Machine Learning on the web is ready");
    while (true) {
        if (knnClassifierModel.getNumClasses() > 0) {
            const img = await webcamInput.capture();

            // Get the activation from mobilenet from the webcam.
            const activation = mobilenetModel.infer(img, 'conv_preds');
            // Get the most likely class and confidences from the classifier module.
            const result = await knnClassifierModel.predictClass(activation);

            //console.log(classes[result.label - 1].name)
            text = classes[result.label - 1].name
            console.log(text)
            predictions.innerHTML = classes[result.label - 1].name
            console.log(result.confidences[result.label])

            confidence.innerHTML = Math.floor(result.confidences[result.label] * 100)

            // Dispose the tensor to release the memory.
            img.dispose();
        }
        await tf.nextFrame();
    }
};
```

Fig 10: Code Example

```
const addDatasetClass = async (classId) => {

    // Capture an image from the web camera.
    const img = await webcamInput.capture();

    // Get the intermediate activation of MobileNet 'conv_preds' and pass that
    // to the KNN classifier.
    const activation = mobilenetModel.infer(img, 'conv_preds');

    // Pass the intermediate activation to the classifier.
    knnClassifierModel.addExample(activation, classId);

    let classIndex = classes.findIndex(el => el.id === classId)
    currentCount = classes[classIndex].count
    currentCount += 1
    classes[classIndex].count = currentCount

    var temp_id = 'images-' + classId.toString()
    document.getElementById(temp_id).innerHTML = currentCount;

    // Dispose the tensor to release the memory.
    img.dispose();
};
```

Fig 12: Code Example

CHAPTER 9

REFERENCES

REFERENCES

These numbers should come in the report in the serial order in which they are being used. Any content from the internet needs to be referenced properly in the manner given below.

[1] https://ieeexplore.ieee.org/

[2] https://www.theatlantic.com/technology/archive/2017/11/

[3] https://techcrunch.com/2014/06/06/

[4] https://lingojam.com/AmericanSignLanguageTranslator

[5] https://economictimes.indiatimes.com/

CHAPTER 10

ACKNOWLEDGEMENT

ACKNOWLEDGEMENT

We would like to express our heartfelt gratitude to our college Fr.Conceicao Rodrigues Institute of Technology for giving us an opportunity to undertake this project. We are grateful to our H.O.D. Mrs Dhanashree Hadsul for giving us a chance to work on this project in our course of Engineering. We are thankful to, and fortunate enough to get constant encouragement, support and guidance from all our teachers of IT Department which helped us in successfully completing our project.
We would also like to extend our gratitude to our parents for their support throughout, and the constant efforts and motivation by our teammates and friends which helped us attain successful completion of this project.