TALKING HAND: A Hand Signal to Speech Converter

Jai Anish Mehta (SBU ID: 114834757)
Department of Computer Science
Stony Brook University
Stony Brook, New York, USA
jaimehta@cs.stonybrook.edu

ABSTRACT

Human-Computer Interaction (HCI) has been an exponentially advancing field and is largely aimed at increasing the efficiency of interaction between humans and computers. This increased efficiency in interaction helps humans to make their lives easier and enhances their comfort. HCI combined with machine learning can be really useful to help people who are physically challenged and are not able to speak. It is observed that people who cannot speak, find it really difficult to have normal conversations with others around them as not everyone is well-trained to understand sign language. As a solution to this issue, I have designed an application called the Talking Hand, which serves as a Hand Signal speech converter. The application takes real-time video input converts it to speech and plays the speech as audio. In this paper, we will learn about the development of the application and how it will be purposeful for physically challenged people.

Keywords: Computer Vision (CV), YOLOv3, Machine Learning (ML), Classification

1. INTRODUCTION

As we move further, humans have always attempted to make the best use of technology to increase the quality of living, thereby increasing comfort. Innovation in technology is brought by human needs and the majority of inventions begin with someone wanting to automate or accomplish a task. As discussed earlier this technology can be served to be really useful for people who are physically challenged. For example, prosthetic robotic limbs are being developed for people who are physically disabled to walk or have challenges in using their limbs. Electronic wheelchairs, hearing aids etcetera are all very good examples of technology being used in this direction. Inspired by the ideas I decided to pursue a project that would help the people who find it challenging or are not able to speak and use hand signals for communication.

People who use sign language for communication, usually use American Sign Language which is a pre-defined set of hand signs dedicated to a comprehensive set of words and commonly referred to as ASL. For the ones who have been trained to use the language in most cases since childhood, it is very easy for them to use the signs to communicate, but when they are communicating with others, they often need a translator who would translate their language for others. Often these translators are the ones from the family who are trained to understand, but not everyone else will be well-trained to understand this language. To eliminate the need for a translator we need a hand-sign language to speech converter that would take hand-sign video as input and convert it into text or speech for others to read or listen to. There exist applications that take the hand-sign image as input and produce output text but there has not been much research into applications that would efficiently convert real-time hand signs into text which would be played as audio for others, additionally, these applications are restricted to the American Sign Language hand-signs and cannot be altered for custom sign languages. Considering the former discussion, I have designed an application that takes real-time video input, performs classification on it using the YOLOv3 [1] (YOLO: You Only Look Once) model weights and produces a video output (as shown in figure 1) along with the hand sign label being played as audio.

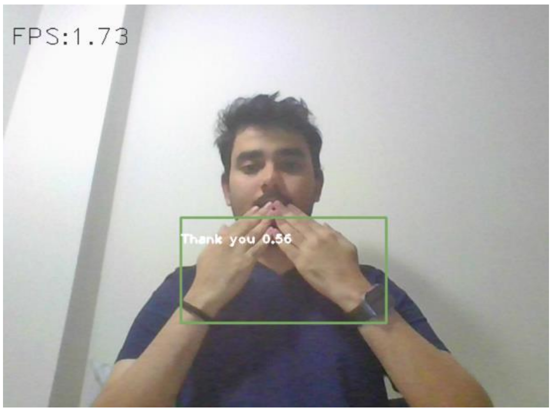


Figure 1: Video output of the application

As we can see in figure 1, where the user has performed the hand-sign thank you and the output shows the bounding box around the hand gesture being detected, with the hand-sign label in the top left corner of the box with the prediction accuracy. Moreover, we can also see the frames per second displayed at the top left corner of the output.

2. MOTIVATION

Inspired by the idea to create an application that can be customised according to the needs of the user I decided to make a hand-sign language to speech converter, that can be personalised for the user. My application can not only record customised hand signals but can also be used to create custom sentences. For demonstrating this functionality, I have trained a class with a custom hand sign labelled “My custom sentence.” Applications that are restricted to ASL can become difficult to use for people who are not trained to use ASL and for that reason they might have to retrain over hand signs. My application solves this issue. Additionally, I have also created the class “Cancel” which can be used to shut the application down making it more convenient.

3. RELATED WORKS

There has been a significant amount of research in the domain of HCI towards the direction of similar applications but they have various different ways of reading the input and classification. There is an application that uses the motion sensors attached to a glove to detect the movements of the hand the person wearing the glove would make. Based on these movements it predicts the hand sign. Additionally, one of the most promising research would be by the well-known company Apple, which uses pulse to detect hand gestures. This detection is done by the sensor on the backside of their widely used product called the Apple watch and then they classify the gestures based on the pulse. At present, it is able to successfully classify gestures like clenching of the fist, snapping of fingers and pinching motion of fingers. In the future apple watch could be used as a hand sign language to speech translator as well. Inspired by these ideas I have tried to create this application that uses YOLOv3 [1] as the classification model. Doing this I try to increase the accuracy of the prediction for hand gestures thereby making the system more accurate. There has not been enough research on making this application integrated with YOLOv3 [1] that would give a speech output.

4. METHODOLOGY

I have divided the approach to the development of the application into majorly five stages. They are as listed below in the order:

i. Data Collection
ii. Data Labelling
iii. Transfer Learning
iv. Classification
v. Text to Speech

The five stages listed above will be discussed in detail in the implementation. Each part builds up on the previous part as we can see that the first two parts are about the data collection and assigning labels to be fed to the model for training. After training classification is performed to produce labels which are finally played as audio. The two major technologies used in order to create the application are given in figure 2.

Figure 2: Technologies used

5. IMPLEMENTATION

Stage I. Data Collection

For this application, I have used my own dataset for two main reasons. Using my own dataset I can personalize the application as I have custom hand signs that I will be labelling and secondly I can optimize the training process by changing the amount of data. To collect the data, I used my laptop camera to capture images of me performing different hand gestures. One advantage of this will be that I will be getting a little noisy data i.e. the images won’t be very high quality and training my model with such data will be better if the user uses a device that may not have a high-quality camera. For the trial, I have collected 20 images per class and for now, I have made the application consisting of six classes that will be discussed in the next stage.

Stage II. Data Labelling

To label the data I have used an application label image [2], which is a python application that takes in image input and allows us to create labels. As you can see in the interface of the application in figure 3, we can select an area we want the algorithm to look at (the blue box I have selected in figure 3).

For illustration, I have used the hand sign “I Love You” in figure 3 around which I have created a bounding box and labelled the image as I Love You which can be seen on the right inside figure 3 in the box labels section. Label Image application has various types of labelling tools to label data according to the requirements of various Machine

Learning algorithms, in our case I have used it for the YOLO [1]. This setting creates a text document for each labelled image as shown in figure 4, containing the class number of the image and the bounding box coordinates that the ML algorithm will use for training purposes. I have created a dataset of 20 images per class, for a total of 6 classes states below:

- Family
- I Love You
- Thank You
- Cold
- My Custom Sentence
- Cancel

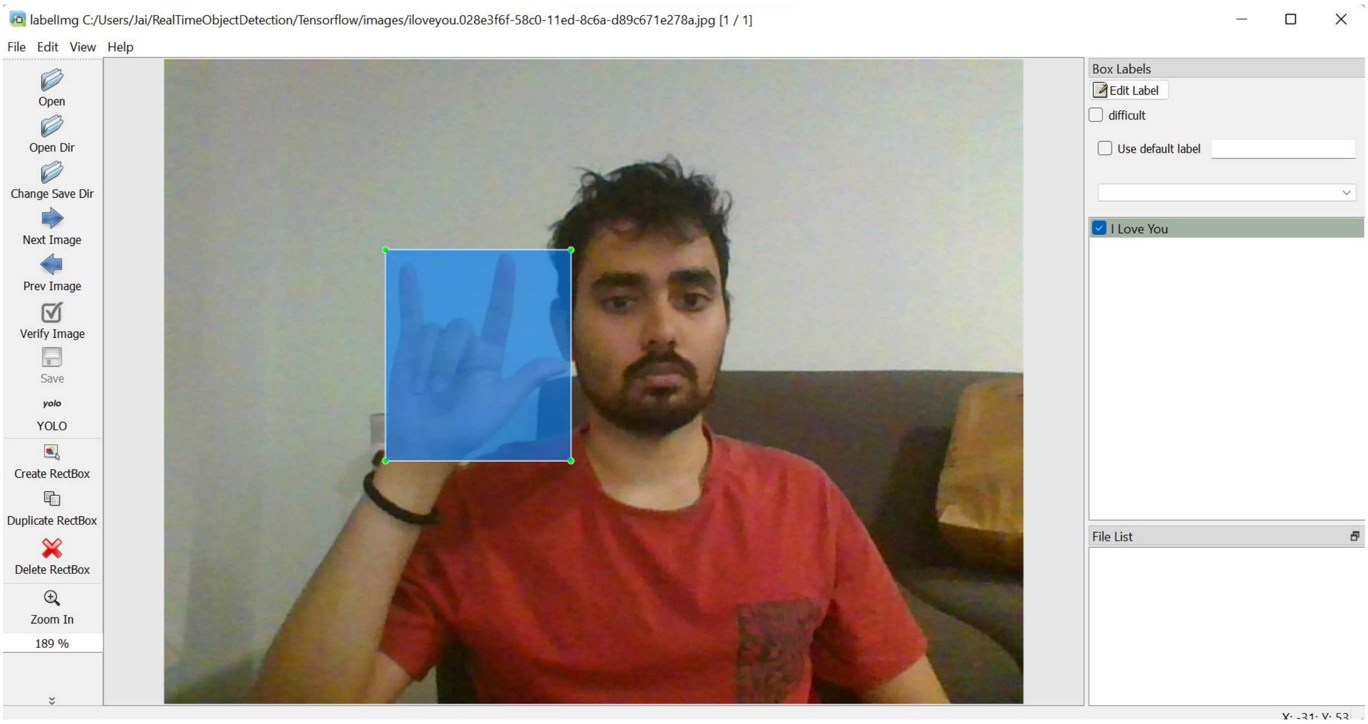
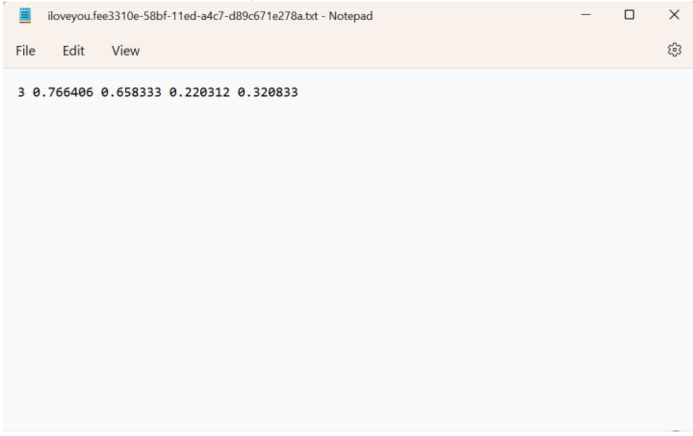


Figure 3: labelImg.py



```
3 0.766406 0.658333 0.220312 0.320833
```

Figure 4: Text file output of the labelled image

Stage III. Transfer Learning

Correctly labelled data and healthy data size is a must for high prediction accuracy, but creating a custom dataset it would require time beyond the scope of the project, hence only 20 images per class have been used. We take a pre-trained model that uses YOLOv3 [1] to predict the presence of humans and cats in an image ad use our dataset to perform transfer learning. For this process we need to change the following variables in the pre-trained mode:

- number of classes
- maximum batches = 2000 * n
- number of filters = (n + 5) * 3

In our case, as we have 6 classes, the maximum number of batches would be 12000 and the number of filters would be 33.

Stage IV. Classification

After performing the transfer learning as stated in stage iii, we get two files that are the last weights that the model used for the classification. We can let the model run as long as we are satisfied with the metrics and once we are satisfied we can stop the model and get the weights file. The downloaded weights file will be used to perform classification on the video input. We take the video input using open cv python and on each frame capture real-time classification is performed to give an output as shown in figure 1 along with the label being converted to speech (will be discussed in the next stage.)

Stage V. Text to Speech

In stage IV we have a label that is generated that we want to be played as speech. For this stage, we use the library pyttsx3 [4], which is a text-to-speech conversion library in Python. I pass the label to the function that would use pyttsx3 [4] to convert it to speech and play it as audio. The best part about the library is that we control the speed of the speech ad we can even add punctuation marks, which can be a part of the future work for this application.

Libraries used:

- cv2
- numpy
- pyttsx3
- glob
- time

6. EVALUATION

This is one of the most important parts of the project as it tells us about the performance of the applications. For evaluating the performance of the application, I choose to perform a heuristic evaluation wherein I chose 10 different users and made them perform all 6 hand signs 5 times each. Hence for each hand sign, we were able to get 30 samples for each hand sign and 300 samples in total. The results of the above evaluation are given in a concise format in table 1.

Class	Number of Correct predictions (out of 30)	Average Prediction Accuracy
Family	23	80%
I Love You	29	89%
Thank You	28	96%
Cold	30	90%
My Custom Sentence	24	78%
Cancel	30	93%

Table 1: Evaluation Results

From the above results, we can see the class that is frequently identified incorrectly (relatively) is “Family”. This is because that hand sign is very similar to that of “Thank You” and hence many times if the hand sign is not performed accurately it may result in incorrect classification. Next would be “My Custom Sentence”, which is a little difficult to perform as it requires the hand sign to be accurately presented in front of the camera. The above results may not seem convincing in all cases but the reason behind the current performance is the amount of training data. With the increase in training data and variation in data, the training process would become more agnostic to different inputs thereby increasing the prediction accuracy.

7. DISCUSSION

The novel concept of the application developed as a part of this project is the customisable hand sign and the hand sign converted to speech. I will be making this dataset available to everyone which can be used for future works on research in this direction. The additional work I would love to pursue on this project is converting a sentence of hand signals into speech with punctuation. As at present each individual label is being converted to speech, having an application that would convert an entire sentence to speech would be really helpful. Additionally, I would like the application to be started by clapping or snapping the finger rather than navigating to the application and running it. Although there is a way to shut down the application with gestures, starting it with a clap would also serve to be more convenient for users.

8. CONCLUSION

We successfully used a combination of technologies like Human-Computer Interaction, Machine Learning and Computer Vision to create a customisable application that would act as hand-signal to speech converter, and would be really helpful for people who are physically challenged and cannot speak. At present the prediction accuracy although seems to be low it is a relatively good accuracy given the amount of data i.e., only 20 images per class for training the model. In future, a higher amount of data that would be rich in quality and amount would be a great contribution towards the project. We used libraries like open cv, pyttsx3 [4] and NumPy majorly to make the development possible. After the evaluation of the application, we can conclude that although the application is performing fairly well, it has a large room for improvement.

10. REFERENCES

[1] https://machinelearningmastery.com/how-to-perform-object-detection-with-yolov3-in-keras/
[2] https://github.com/heartexlabs/labelImg
[3] https://github.com/AlexeyAB/darknet
[4] https://pypi.org/project/pyttsx3/