IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Keyboard based(Static) American Sign Language Communication- *KASLC* System

Jatin Saluja, Luv Karanwal, Sanket Gupta
Department of Computer Science & Engineering
Medicaps University, Pigdambar(Rau), Indore(Madhya Pradesh)

Sign language has always been considered a trivial mode of communication since ancient times which prevails now as a hobby or a mode of communication for people with disabilities of speaking or Hearing Impairments. Sign Language is the ascendant yet non-primary form of the communication language used in large groups of people in society. According to the World Health Organization (WHO) report in 2020, there are more than 466 million deaf people in the world. It's safe to say that Sign Language is a complete and natural language, that conveys information through hand/arm gestures, facial expressions, head movements, and body postures.

Sign Language is the prevalent yet non-underlying form of communication language used in the mute, deaf, voice impaired and hearing impaired community. Sign Language is a complete and natural language that conveys information through hand/arm gestures, facial expressions, head movements, and body postures. To make an easy and mutual communication between the vocal impaired and the hearing communities, building a robust system capable of translating the sign language into spoken language and vice versa is fundamental. In the system proposed, sign language recognition and production are two significant parts for making a two-way communication system. Sign language recognition needs to cope with some critical challenges. In this Project, we tried to advance Sign Language Communication using Machine learning. We aim to optimise the highly expensive conversion process of translating sign language to spoken language, using gloves or high end devices to a simple Cloud and webcam based process.

1. Introduction

Keyboard based(Static) American Sign Language Communication System is our effort to make life easier for them and make us understand them better via communication.Keyboard based(Static) American Sign Language Communication System Detects static ASL Hand Gestures and Converts them to Voice via Transcription.

The system developed is a Python based Project working on Machine Learning algorithms of TensorFlow library. It is trained with Modified counters based black and white Datasets to make it work on all colours of plain backgrounds and with all sorts of Skin Colors and Hand Sizes. The Approach we used is Detection Region based prediction model trained on 700+ training images and 40+ testing images, accumulated with the help of a python OpenCV library.
The System is optimised to work on different colour background conditions with a DataSet Optimisation for faster processing. We've used the counters based approach for optimising the system. The Approach eliminates the role colors from the Images and distinguishes foreground and background hence making it more traceable for multiple color backgrounds and Skin Colors.

2. Why KASLC?

Sign Language as a tool of Communication isn't known to everyone which makes the disables feel secluded and be treated differently from the normal society. To fulfill that communication gap and make non verbal communication more feasible between both sections of society.

Our Primary Objectives while developing the project were:

a. To offer a cheap communication system between Voice Impaired and a Verbally Active community.
b. Form an integration model acting as a bridge between Sign Language and Verbal Languages
c. Learn in depth about Machine Learning Tools and different Models
d. Learn Cloud Deployment and Integration Technologies.

3. Implementation Method

Any user trying to implement the system is required to fulfill the following Hardware and Software Requirements:

- An End Device
- Active Internet
- Python 3.0+
- Jupyter Notebook
- OpenCV Python module for video capturing
- PIL Python Module for Image Processing
- OS-Win Python module for Accessing Path Variables
- Time Python module for managing processes at a particular time
- UID Python Module for automatically generating distinct file names with specific labels.

The system can be implemented as 3 major components :

A. **Dataset Accumulator:** In this component_We will be taking a live feed from the webcam and every frame that detects a hand in the detection region (Detection Box) created will be saved in a gesture directory that contains two folders train and test, containing folders containing images captured using the data_accumulator.py.

For creating the dataset we get the live cam feed using OpenCV and create an Detection Region that is nothing but the part of the frame where we want the hand to be in while detecting the hand for the gestures.
The red/blue box is the Detection Region and current window is for getting the live cam feed from the webcam.
Then, for differentiating between the background we calculate the accumulated weighted avg for the background and then subtract it from the frames that contain some object in front of the background to distinguish as foreground. This was done by calculating the accumulated_weight for some frames. After we got the accumulated avg for the background, we subtracted it from every frame we read after 60 frames to find any object that covers the background.
We put up a text using cv2.putText to display "Fetching Background" in the Detection Region while detecting the background so as to remind the user to avoid any foreground insertion while calculating the accumulated_weight.
Then we calculated the threshold value for every frame and determined the contours using cv2.findContours and returned the max contours (the most outermost contours for the object) using the function *segment*. Using the contours we were able to determine if there is a hand in the Detection Region ie. if there is any foreground object being detected in the Detection Region.
When a hand is present in the Detection Region(ie. contours are detected), We start to save the image of the Detection Region in the train and test set respectively for the characters we want it to be detected as.

For this project, we saved 701 images for each of the 29 characters and notions to be detected,in the train dataset and for the test dataset, we did the same and created 40 images for each notion.

B. **Training the CNN:** With the accumulated Data, we started to Train the CNN.

First, we load the data using the ImageDataGenerator of keras API through which we can use the flow_from_directory function to load the train and test set data, and each of the names of the dataset folders will be the class names for the images loaded.
Used plotImages function for plotting images of the dataset loaded.
In training callbacks of Reduce LR on plateau, early_stopping is used, and both of them are related to validation dataset loss.
After every stage, the accuracy and losses were calculated using the validation dataset and if the validation loss is not decreasing, the LR of the model is reduced using the Reduce LR to prevent the model from overshooting the minima of loss and also we are using the early_stopping algorithm so that if the validation accuracy keeps on decreasing for some stages then the training is stopped.
The train_model.py contains the callbacks used, also it contains the two different optimization algorithms:

A. *SGD (stochastic gradient descent)*- The weights are updated at every training instance
B. *Adam*- combination of Adagrad and RMSProp.

We found for the model SGD seemed to give higher accuracies.
As we observed while training we found 100% training accuracy and validation accuracy was about 81%
After compiling the model we fit the model on the train batches for 10 stages or epochs (can vary according to the choice of parameters of the user), using the callbacks discussed above.We then got the next batch of images from the test data & evaluating the model on the test set and printing the accuracy and loss scores. Here we visualised and made a small test on the model to check if everything is working as we expect it to while detecting on the live cam feed.

A dictionary is then introduced with label names for the various labels predicted as word_dictionary.

C. **Gesture Detection:** In Use_Model.py, we created a bounding box to act as Detection Region and calculated the accumulated_avg as we did in the Dataset Accumulator for identifying any foreground object.
Then we found the max contour and if contour is detected that means a hand is detected so the threshold of the Detection Region is treated as a test image.
We load the detection_model.h5 using keras.models.load_model and feed the threshold image of the Detection Region consisting of the hand as an input to the model for prediction.

The detection model can further be deployed to cloud services and can be accessed with any end device with an internet connection and a camera using a device compatible software corresponding to Use_Model.py

The system has 2 types of output:

a. **Intermediate or Detection-*Time Output:*** The system transcribe the characters defined by each gesture in runtime within a fraction of seconds as defined in Use_Model.py
b. **Final Output:** The final output of the system is a voice based output reading the transcribed set of characters as words. It has to be initialised with a pre-defined gesture.

4. Related work

A lot of models are based on sign language detection but a lot of them either use Large Datasets, Requires high end Machine Learning Tools, Restricts Gestures or Can't detect certain Backgrounds. Some of the references and their limitations are listed Below:

Project Name	**Limitations**
hthuwal / sign-language-gesture-recognition	Large Video based Dataset
loicmarie / sign-language-alphabet-recognizer	Failed to capture Image
harshbg / Sign-Language-Interpreter-using-Deep-Learning	Gesture Restrictions
0aqz0 / SLR	Detection Time and Isolation of words
rrupeshh / Simple-Sign-Language-Detector	Insufficient Data Set
soumik12345 / Kinect-Vision	Works only for Games

The major methods in the model are inspired from Dataflair's ML training model for detecting the number of fingers and representing the count on screen. The system is trained on a dataset based on gestures generalised as American Sign Language(ASL) with a slight variation in characters "M" and "N".
The variation was made as the proposed methods in the system can't distinguish between them due to gestures complexity.

5. Scope of the Proposed Work

The Model is an innovative yet cheap way to offer a way of communication between Voice Impaired people and Verbally Active people with a simple camera and Internet enabled machine. The model with the help of proper Cloud integrations can easily act like a bridge between Sign Language and Verbal Languages. With the help of more variable(larger yet optimised) Dataset integrations and interactive UI Applications integrations it can play a crucial role in facilitating a lot of Commercial Sign Language Projects by being a key component or a major method.

6. Conclusion

The system is trained on a dataset created based on gestures generalised as American Sign Language(ASL) with a slight variation in characters "M" and "N".
The variation was made as the proposed methods in the system can't distinguish between them due to gestures complexity.
The major problems encountered were while training the model. They were due the long training time, caused due to large datasets and unavailability of GPU resources. The patience required to deal with this problem can easily be eliminated by using High End devices or Cloud based ML resources.
The system can't detect gestures while using backgrounds with prints or textures, there's a problem in detection due to low lighting conditions.

References:

[1] .R. Y. Wang, J. Popovi´c, "Real-time hand-tracking with a color glove", ACM Transactions on Graphics 28 (3) (2009) 63. 13.

[2] R. Alzohairi, R. Alghonaim, W. Alshehri, S. Aloqeely, M. Alzaidan, O.Bchir, "Image based arabic sign language recognition"International Journal of Advanced Computer Science and Applications, Vol. 9, No. 3, 2018. 14.

[3] J. Zieren, K.-F. Kraiss, "Robust person-independent visual sign language recognition," Conference on Pattern Recognition and Image Analysis, 2005, pp. 520–528. 15.

[4] T. Starner, J. Weaver, A. Pentland, "Real-time American sign language recognition using desk and wearable computer based video", IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (12) (1998) 1371–1375. 16.

[5] J. Rekha, J. Bhattacharya, S. Majumder, "Shape, texture and local movement hand gesture features for Indian sign language recognition",3rd International Conference on Trendz in Information Sciences & Computing, 2011, pp. 30–35. 17.

[6] N. Tanibata, N. Shimada, Y. Shirai, "Extraction of hand features for recognition of sign language words", International conference on vision interface, 2002, pp. 391–398. 18.

[7] F.-S. Chen, C.-M. Fu, C.-L. Huang, Hand gesture recognition using a real-time tracking method and hidden markov models, Image and vision computing 21 (8) (2003) 745–758. 19.

[8] M. A. Mohandes, "Recognition of two-handed Arabic signs using the cyber glove," Arabian Journal for Science and Engineering, vol. 38, no. 3, pp. 669–677, 2013. 20.

[9] X. Zhang, X. Chen, Y. Li, V. Lantz, K. Wang, J. Yang, "A framework for hand gesture recognition based on accelerometer and EMG sensors", IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans 41 (6) (2011) 1064–1076. 21.

[10] N. Tubaiz, T. Shanableh, K. Assaleh, "Glove-based continuous Arabic sign language recognition in user-dependent mode", IEEE Transactions on Human-Machine Systems 45 (4) (2015) 526–533.

[11] R. A. Kadry and A. Birry, "ASL Recognition Using Leap Motion and Hand Tracking Mechanism," Int. J. Adv. Electron. Comput. Sci., vol. 4, no. 9, pp. 2393–2835, 2017. 39.

[12] L. E. Potter, J. Araullo, L. Carter, "The leap motion controller: a view on sign language",25th Australian Computer-Human Interaction Conference: Augmentation, Application, Innovation, Collaboration, 2013, pp. 175–178. 40.

[13] M. Mohandes, S. Aliyu, and M. Deriche, "Arabic Sign Language Recognition using the Leap Motion Controller". June, 2014. 41.

[14] M.Funasaka, Y.Ishikawa, M.Takata and K.Joe," Sign Language Recognition using Leap Motion Controller",2015, pp. 263-269. 42.

[15] A. Elons, M. Ahmed, H. Shedid, M. Tolba, "Arabic sign language recognition using leap motion sensor," 9th International Conference on Computer Engineering & Systems, 2014, pp. 368–373. 43.

[16] N. B. Ibrahim, M. M. Selim, and H. H. Zayed, "An Automatic Arabic Sign Language Recognition System (ArSLRS)," J. King Saud Univ. - Comput. Inf. Sci., 2017. 44.

[17] Amanda Duarte. Cross-modal neural sign language translation. The 27th ACM International Conference, 2019. 6

[18] Amanda Duarte, Shruti Palaskar, Deepti Ghadiyaram, Kenneth DeHaan, Florian Metze, Jordi Torres, and Xavier

[19] GiroiNieto1. How2sign: A large-scale multimodal dataset or continuous american sign language. Sign Language Recognition, Translation, and Production workshop, 2020.

[20] Sarah Ebling and Matt Huenerfauth. Bridging the gap between sign language machine translation and sign language animation using sequence classification. Proceedings of SLPAT 2015: 6th Workshop on Speech and Language Processing for Assistive Technologies, pages 2–9, 2015. 4

[21] Sarah EblingJohn and GlauertJohn Glauert. Exploiting the full potential of jasigning to build an avatar signing train announcements. In 3rd International symposium on sign language translation and avatar technology, pages 1–9, 2013.

[22] Eleni Efthimiou, Stavroula-Evita Fotinea, Thomas Hanke, John Glauert, Richard Bowden, Annelies Braffort, Christophe Collet, Petros Maragos, and Franc¸ois

[23] Lefebvre-Albaret. The dicta-sign wiki: Enabling web communication for the deaf. International Conference on Computers for Handicapped Persons (ICCHP), pages 205–212, 2012.
[24] Ethnologue. Argentine sign language.
https://www.ethnologue.com/language/aed, 2021.
[25] Ethnologue. Greek sign language.
https://www.ethnologue.com/language/gss, 2021.
[26] Ethnologue. Persian sign language.
https://www.ethnologue.com/language/psc, 2021.
[27] Jens Forster, Christoph Schmidt, Thomas Hoyoux, Oscar Koller, Uwe Zelle, Justus Piater, and Hermann Ney. Rwthphoenix-weather: A large vocabulary sign language recognition and translation corpus. Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC12), Istanbul, Turkey, page 3785–3789, 2012.
[28] Sakher Ghanem, Christopher Conly, and Vassilis Athitsos. A survey on sign language recognition using smartphones. Proceedings of the 10th international conference on pervasive technologies related to assistive environments, Island of Rhodes Greece, 2017.
[29] Owlcation. Korean sign language. https://owlcation.com/humanities/Korean-Sign-Language, 2021.
[30] Siegmund Prillwitz. Hamnosys. version 2.0. hamburg notation system for sign languages. an introductory guide. Hamburg Signum Press, 1989.
[31] Razieh Rastgoo, Kourosh Kiani, and Sergio Escalera. Multimodal deep hand sign language recognition in still images using restricted boltzmann machines. Entropy, 20, 2018.
[32] Razieh Rastgoo, Kourosh Kiani, and Sergio Escalera. Hand sign language recognition using a multi-view hand skeleton. Expert Systems With Applications, 150, 2020.
[33] Razieh Rastgoo, Kourosh Kiani, and Sergio Escalera. Videobased isolated hand sign language recognition using a deep cascaded model. Multimedia Tools And Applications, 79:22965–22987, 2020.
[34] Razieh Rastgoo, Kourosh Kiani, and Sergio Escalera. Hand pose aware multimodal isolated sign language recognition. Multimedia Tools And Applications, 80:127–163, 2021.
[35] https://arxiv.org/pdf/2103.15910v1.pdf
[36] https://data-flair.training/blogs/sign-language-recognition-python-ml-opencv