

Real Time Conversion of Sign Language using Deep Learning for Programming Basics

Thanasekhar B
thanasekhar@gmail.com
Department of Computer Technology
Anna University, Chennai-600044

Akshay V
akshayred@gmail.com
Department of Computer Technology
Anna University, Chennai-600044

Deepak Kumar G
deepakcr974@gmail.com
Department of Computer Technology
Anna University, Chennai-600044

Abdul Majeed Ashfaaq
mamashfaaq@gmail.com
Department of Computer Technology
Anna University, Chennai-600044

Abstract—Sign Languages are languages that use the visual-manual modality to convey meaning. It serves as a medium of communication between speech-impaired people and normal people. They are full-fledged natural languages with their own grammar, lexicon and classifications such as American, Indian, Chinese and so on. However, due to the non-universality of sign languages, there is a huge communication gap between the speech or hearing-impaired people and others. Hence, they also suffer from access to basic education. In this work, a low-latency real-time sign language recognition application is developed for detecting and processing gestures performed from the Indian Sign Language (ISL) dictionary using a Convolutional Neural Network model, and identify the words that are being communicated. We focus our application on words from a specific domain, i.e., basic programming keywords, through a custom-made dataset containing 500 different images of the gesture corresponding to each word. Our application strives to detect both static and dynamic gestures performed by the user, and generate Python-like syntax for various constructs such as if and loop, with a minimal response time of less than 0.1s. This can potentially make inroads into educating disabled people in the field of programming.

Keywords : Sign Language, gestures, CNN, programming basics, disabled people

I. INTRODUCTION

Sign Languages break the barrier for communication for speech and hearing-impaired people. To design technological systems that can interpret sign language, one can combine the potential of computer vision and image recognition frameworks. Sign Languages are languages that use the visual-manual modality to convey meaning and to communicate. They are full-fledged natural languages with their own grammar and lexicon. It is generally classified as American, Indian, Chinese and so on, depending on the region of usage. This non-universality of sign language creates a communication gap between normal people and the colloquial users of the sign language. In March 2018, an ISL (Indian Sign Language) [5] dictionary was officially released by the Union Ministry, comprising of 3000 words, creating scope for national standardization of the sign language. The primary reason behind such an initiative was to encourage the language and standardize the medium for communication.

Some of the challenges involving sign language are,

- People think in terms of direct English words for communication, while sign language requires them to use

abstraction and other skills to communicate effectively.

- Deficit of sign language interpreters available, for communication between normal people and the impaired ones.
- The education sector, for one, is in immediate need for intervention for the benefit of the disabled people. The resources to learn from are very sparse, and this has hindered their development and growth.

For example, learning programming, which is a much-needed skill in today's technologically driven world, is very difficult for hearing impaired people.

Gesture recognition is an open problem in the area of machine vision, a field of computer science that enables systems to emulate human vision. Gesture recognition has many applications in improving human-computer interaction, and one of them is in the field of Sign Language Translation, wherein a video sequence of symbolic hand gestures is translated into natural language. Gestures performed as a part of sign language can be complex to the naked eye, due to a great amount of distinguishable details. This creates good scope for image classification and machine learning algorithms to tap into, by classifying gestures performed and identifying the most probable gesture. Sign Languages use manual gestures for communication. With regional ISL dialects and dictionaries, there is a pressing need for autonomous systems that can understand and translate sign language gestures to a universally understandable format. The prediction of gestures using computer vision and deep learning techniques can run into several bottlenecks, such as distortion and noise in the input image frame, and detection of the hand region within the image frame. Both these cases need to be handled efficiently for the model to perform optimally in a real-time prediction environment. The gist of the proposed solution is as follows,

- Combining the power of computer vision and deep learning algorithms such as CNN (Convolutional Neural Networks) which are tailor made for image classification, one can develop systems that bridge the gap between hearing, speech impaired people and others.
- Programming, which is a relevant skill in today's rapidly developing world, is virtually inaccessible to hearing-impaired people by traditional teaching methodologies. However, a simple workaround is that it can be taught through gestures corresponding to the

keywords in the coding languages.

- The proposed system can be leveraged to help hearing impaired people to learn programming, using the power of technology.

II. RELATED WORK

A. Hand Detection

Chaudhary et al. [1] divide hand detection into two parts: appearance based and a model-based approach. In appearance-based approaches, fingertips are detected to enable hand-gesture recognition. The approach uses a neural network-based system that recognizes continuous hand postures from gray level video images. Conversely, in the model-based approach, they use a histogram for calculating the probability for skin color observation.

Raheja et al., [11] proposes a new methodology for real time robot control using Principal Component Analysis (PCA) for gesture extraction and pattern recognition with saved images in database in 60x80 image pixels formats. The author uses syntax of few gestures and decides corresponding actions of robot, and claims that PCA method is comparatively faster than neural network-based methods which require training database and higher computation power.

B. Static Gesture Recognition

Sanil et al., [13] predict Sign Language characters such as alphabets and numbers on the basis of the gesture performed. The author uniquely takes care of both single-handed and double-handed gestures. With a manually collected dataset, they extract both SIFT (Scale Invariant Feature Transform) keys as well as HOG (Histogram of Oriented Gradients) features from images for training purposes, with varying accuracies. Further, hierarchical classification is performed by initially checking for one handed-two handed gestures and then predicting the actual alphabet, for improved accuracies. The SIFT features yield an accuracy of 32.74%, while HOG features combined with Hierarchical classifier yields 53.23%.

Sakshi et al., [12] extract SIFT features to perform gesture prediction. SIFT features help in detecting corners of images irrespective of scaling. The model separates the system into four modules: Image Acquisition, Feature Extraction, Orientation Detection and Gesture Recognition. In the feature extraction step, several submodules vis-à-vis Scale Space Extrema Detection, Keypoint Localization, Orientation Assignment, Keypoint descriptor are devised.

Neha et al., [10] captures gesture images and recognize a system of alphabets and numerals. The HOG features are extracted and passed to a neural network for the gesture recognition purpose. Artificial Neural Networks are known to train themselves best in the domain of image classification and this is simulated in order to perform optimal character recognition.

Ghotkar et al., [4] ideate a system consisting of four modules, vis-a-vis real time hand tracking, hand

segmentation, feature extraction and gesture recognition. Camshift method and Hue, Saturation, Intensity (HSV (Hue Saturation Value)) color models are used for hand tracking and segmentation. For gesture recognition and prediction, Genetic Algorithm is used. A standard procedure of Crossover, Mutation and Replacement is followed until the best solution is found.

C. Dynamic Gesture Recognition

Abid et al., [9] formulate a way to employ dynamic hand gestures for controlling smart home interactive applications. A 3D multitask part model is used for event ordering mechanism. A linear grammar is used to validate and recover from erroneous commands. The system is divided into IP module and SLFG (Stochastic Linear Formal Grammar) module. For training purposes, 16 scenarios are considered while generating the manual database, with varying positions of the hand, distance between the hand and the camera, background and angle of inclination with respect to the camera lens. The BOF (Bag of Features) model is used to extract features from the video sequence, after reducing the size of each video using down sampling. The video features are encoded using a multiscale scanning window. HOG3D, a form of 3-D oriented gradient is computed using an integral video method, to generate spatial-temporal gradient histograms. For classification, the approach uses BOF and non-linear SVM (Support Vector Machines).

Joslin et al., [2] use a dynamic hand-gesture recognition method. Their focus is on tracking fingers and hands to attain information about gestures. They use a camera-based system to record information about the gesture, and this is processed from a 2D space to 3D space to obtain a broader understanding of the gesture. Orientation cues and hand constraints are employed during the 2D to 3D conversion. However, their technique is low-speed and inaccurate. The authors discuss influence of specific contexts over a recognized gesture, such as environmental context, situational context and user context.

D. CNN for Gesture Recognition

Anantha Rao et al., [3] propose a CNN based Sign Language Recognition system for gesture prediction. A custom-dataset with 5 subjects performing 200 different signs in different angles is procured for this purpose. 4-layer convolution along with stochastic pooling are observed to produce optimal results, and they are compared with ANN (Artificial Neural Network) and Adaboost. Low level features like lines, edges and corners are learned from convolutional layer 1 and 2. Higher level features corresponding to the images are learned from convolutional layer 3 and 4.

This method produces an accuracy of 92.88% with stochastic pooling method. With max pooling and mean pooling, accuracies of 91.33% and 89.84% are observed. This work also compares this performance of CNN with other classification methods such as Mahalanobis distance classifier (MDC), ANN and deep Artificial Neural Networks. Though CNNs take more time for training, the testing comparatively takes far less computation times.

Rashedul et al., [8] train a Deep Convolutional Neural Network (DCNN) on the American Sign Language alphabet dataset. The authors use a bounding box to separate the region of the interest in the image. They also employ background subtraction to cancel noise and convert input images from 3- channel to 1-channel grayscale to down-sample them.

From the background study, it has been observed that very little research has gone into Indian Sign Language recognition and assistance systems. It has also been inferred that Con- volutional Neural Networks can be utilized to detect static and dynamic gestures corresponding to sign language, and combining this with the utility of computer vision frameworks, one can develop systems that aide disabled people.

III. PROPOSED WORK

Though sign language has been studied for quite some time, most of the work has been focused on finger-spelling. Alphabet recognition and numeral recognition have provided scope for many computer science enthusiasts to develop systems that de- tect and predict finger-spelling gestures. Static Gesture Recog- nition focuses on such finger spelling gestures usually. How- ever, majority of the communication in sign-language takes place through word-level gestures, which are very dynamic in nature. These require systems that can process image frames in real-time and analyze and predict gestures that have been performed.

We propose to create a real-time sign language recognition system using image processing, computer vision and deep learning algorithms, that is capable of detecting and predicting gestures from the ISL dictionary dynamically. In order to direct our efforts towards a more application-oriented project, we create a programming syntax generator based on sign language gestures. Historically, education has not been easy for the disabled people. Our sign language model is specifically restricted to the programming language keywords domain, and mainly highlights the syntaxes different programming constructs in pseudocode terms. This can be considered as an avenue to continue further work on, where hearing-impaired and speech-impaired people can acquaint themselves with the world of programming languages.

A. Frameworks Used

In programming syntax generation phase, we work with 7 gestures Begin, Condition, Else, End, If, Loop and Statement. Of these, 4 gestures are dynamic, vis--vis Begin, Condition, Statement and Loop. Dynamic gestures are those which cannot be uniquely identified by studying a single frame. A custom- made dataset is once again prepared for these gestures. The architecture of this system is depicted in Figure 1. Several CNN models were trained using this gestures dataset which had incremental increase in accuracies and prediction speeds. A greedy algorithm is used to predict only necessary frames and reduce redundant predictions. Finally, a syntax generation module is employed to display the keywords pre- dicted in a neat Python-like syntax. The model is capable of generating code corresponding to if and loop constructs. We replace

braces with the begin and end gestures for improving readability.

Image processing frameworks such as cv2, OpenCV and pillow are used for hand-detection, skin segmentation and cropping tasks during the different phases of the project. To train the model, we use Deep Learning and ML frameworks such as keras [6] to train on Convolutional Neural Networks while using scikit-learn [14] library for other algorithms such as Decision Tree and Support Vector Machines. For the UI part of the application, kivy [7] framework is used.

B. Architectural Diagram

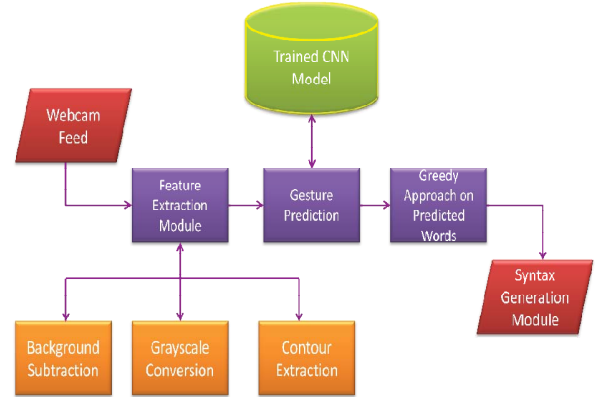


Fig. 1. Sign Language to Programming Syntax Architecture

C. Dataset Used

Since Indian Sign Language (ISL) gesture recognition is a less explored topic we couldn't find the data needed to accurately train CNN models. So we had to create a dataset on our own. The dataset consists of 400 images each of the ISL signs for the programming language keywords: Begin, End, If, Else, Condition, Statement, Loop of which Begin, Condition, Statement and Loop are dynamic gestures.

Gesture	Prediction
	Else
	End
	If

Fig. 2 Static Gestures

D. Algorithm

In order to create the dynamic gesture dataset where each

gesture is a video, frames were captured at the rate of five per sec as the gesture was performed. All the frames corresponding to a gesture were given the same label so that the model can learn to generalise between the different frames involved in a gesture. It is essentially a Many-to-One gesture prediction where for static gestures it is One-to-One.

Another point worth noting is that If and Else did not have specific gestures in the ISL dictionary, and hence they were taken from the ASL (American Sign Language) dictionary which had unique gestures for the aforementioned words. Figures 2 and 3 depicts the gestures corresponding to the words.

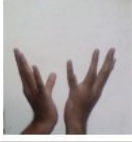
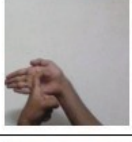


Gesture	Prediction
	Begin (Dynamic)
	Condition (Dynamic)
	Statement (Dynamic)
	Loop (Dynamic)

Fig. 3 Dynamic Gestures

```

begin
  if
    condition:
      begin
        statement;
        loop
          condition:
            begin
              loop
                condition:
                  begin
                    statement;
                    statement;
                  end
                end
              end
            end
          end
        end
      end
    else:
      begin
        statement;
      end
    end
  end
end

```

Fig. 4 Syntax Generated

An interval of two seconds is set for a single gesture identification. The fps is set to 10. Each frame is sent for

prediction and the count of the predicted gesture is incremented. At the end of two seconds a total of 20 frames will be processed and the gesture which has been predicted the most is sent to syntax generation module. Algorithm 1 contains the pseudo-code for this method.

E. Syntax Generation

For generating the relevant syntax, the following points are to be noted:

1. Python-style syntax is followed.
2. Begin and End gestures are used as a substitute for braces.
3. After each begin gesture the indentation weight is incremented so that all the gestures following it will be within the same block. Once end gesture is predicted the indent weight is decremented so that the future gestures will belong to the previous block.

Algorithm 1: Gesture prediction from user actions

```

predCount[] = [0,0,0,0,0,0,0]
numFrames = 0
while camera.isOpen() do
  predProb = predict(model, currFrame).prob
  predClass = predict(model, currFrame).class
  if predProb*100 >= 85 and numFrames <= 20
    then
      predCount[predClass]+=1
    end
  if numFrames == 20
    then
      finalPrediction = max(predCount).index
      predCount[] = [0,0,0,0,0,0,0]
      numFrames = 0
    end
  numFrames+=1
end

```

No. of Convolution layers	No. of MaxPool layers	No. of Dense layers	No. of Dropout layers	Accuracy (%)
4	3	2	1	33
3	3	3	2	61.58

Fig. 5 CNN Model Architecture and Accuracy

IV. PERFORMANCE METRICS

The model was trained on 2800 images from the custom English-words dataset on Convolutional Neural Networks(CNN) with Ada-boost optimizer. A Kaggle kernel with 14 GB RAM and GPU (Graphics Processing Unit) support is used for evaluation purposes, with 700 images corresponding to the 7 different classes used as

test-set. The frames are captured from a Dell Inspiron 5559 Webcam with 0.92MP camera. The training takes place for 50 epochs. An important aspect of this dataset was that 4 of the 7 classes had dynamic gestures.

The dynamic gesture dataset of 7 classes with each class having 400 training and 100 test image data, performed moderately well for a CNN architecture with 3 convolutional layers and 3 maxpool layers, with an accuracy of 61.58% as depicted in Figure 5.

V. CONCLUSION

With this application, we have created a proper Indian Sign Language Recognition system that can deal with both static and dynamic gestures, with an average accuracy of 1.58%. This system is as close to real-time as possible, with optimizations for speed wherever necessary. This system also handles one-handed and two-handed gestures within the same model. Having delved into this project with very little work to refer to, due to the relative negligence towards ISL by researchers, we have made significant inroads into making a system that can deal with ISL gestures. It is also observed that a diverse and extravagant dataset with more variations could make the model robust to sign variations, and improve the results significantly.

By focusing this system towards programming basics, we have introduced a novel method to take programming knowledge to the disabled of people all over the world, and improving access to education for all.

VI. FUTURE WORK

Future work can concentrate on improving the scope for dynamic gesture recognition. ISL majorly has dynamic gestures, some of them quite similar to each other, so with more data and algorithms, a system that can independently interpret sign language gestures and assist the users can be created. Over 1.3 million people are classified as hearing impaired and 1.6 million people are classified as speech impaired in India, according to the 2011 Census. This creates a pressing need for technology to intervene and solve the issues of the disabled people.

These systems can also concentrate on specific domains, similar to this work, i.e., programming syntax keywords. Taking education and knowledge to the masses is important, and disabled people can benefit greatly from the systems that can help tutor them. Programming, being a must-know skill in today's technologically driven world, should be accessible to all.

In this system, predictions are displayed based on a greedy algorithm. Though this algorithm works well for a real-time prototype application, a wide-spread user system will require much more advanced gesture intervals identification techniques to determine when they are performed. One can perform additional machine learning routines to determine when the gesture predictions need to be displayed. There is a need for robust systems that can

work with just the hand region wherever it is present (without a bounding box), and whatever may be present behind it.

One of the most important components of machine learning projects is data. With more diverse and better quality of datasets, these sign language prediction systems can become more robust and powerful, which would eliminate the requirement for sign language interpreters. More the data, greater the computation power required for entities such as neural network to learn from them. With advanced GPU systems available at comparatively lower costs, the scenario is bright for advanced systems. Autonomous systems that can make life easier for speech-impaired and hearing-impaired people can go a long way in solving some of humanity's biggest challenges.

REFERENCES

- [1] A. Chaudhary, J. L. Raheja, K. Das, and S. Raheja, "A survey on hand gesture recognition in context of soft computing", in *Proc. CCSIT*, vol. 133. Jan. 2011, pp. 46 - 55.
- [2] Joslin, A. El-Sawah, Q. Chen, and N. Georganas, Dynamic gesture recognition, *IEEE transactions on IMTC*, May 2005, pp. 1706 - 1711.
- [3] G. Anantha Rao, K. Syamala, P. V. V. Kishore, A. S. C. S. Sastry, Deep Convolutional Neural Networks for Sign language recognition, *Conference on Signal Processing and Communication Engineering Systems*, Jan. 2018, pp. 851 - 856.
- [4] Ghotkar, Archana S., Hand Gesture Recognition for Indian Sign Language, *International Conference on Computer Communication and Informatics (ICCCI)*, 2012, pp. 1 - 4.
- [5] Indian Sign Language Dictionary Press Release by Union Ministry, <http://pib.nic.in/newsite/PrintRelease.aspx?relid=177900>, last accessed on: Apr. 4, 2019.
- [6] Keras: The Python Deep Learning Library, <https://keras.io/>, last accessed on: Apr. 4, 2019.
- [7] Kivy: Open Source Library for Python User Interface, <https://kivy.org/#home>, last accessed on: Apr. 4, 2019.
- [8] Md. Rashedul Islam, Ummey Kulsum Mitu, Rasel Ahmed Bhuiyan, Jungpil Shin, Hand Gesture Feature Extraction Using Deep Convolutional Neural Network for Recognizing American Sign Language, *International Conference on Frontiers of Signal Processing*, Sep. 2018.
- [9] Muhammad Rizwan Abid, Emil M. Petriu, Ehsan Amjadian, Dynamic Sign Language Recognition for Smart Home Interactive Application Using Stochastic Linear Formal Grammar, *IEEE Transactions on Instrumentation and Management*, 2015, pp. 596 - 605.
- [10] Neha V. Tafari, P. A. V. D. Indian sign language recognition based on histograms of oriented gradient, *International Journal of Computer Science and Information Technologies*, Oct. 2014, pp. 3657- 3660.
- [11] Raheja J. L., Shyam R. Kumar, U. Prasad, Real-Time

Robotic Hand Control using Hand Gesture, 2nd International conference on Machine Learning and Computing, Bangalore, India, Feb. 2010, pp. 12 - 16.

[12] Sakshi Goyal, Ishita Sharma, Shanu Sharma, Sign Language Recognition System for Deaf and Dumb People, International Journal of Engineering Research and Technology (IJERT), Oct. 2013, pp. 382 - 387.

[13] Sanil Jain, K.V.Sameer Raja, Indian Sign Language and Character Recognition, <https://cse.iitk.ac.in/users/cs365/2015/submissions/vinsam/report.pdf>, last accessed on: Apr. 4, 2019.

[14] Scikit-learn: Machine Learning in Python, <https://scikit-learn.org/stable/>, last accessed on: Apr. 4 2019.