

DACTYLOLOGY RECOGNITION SYSTEM

Aleti Meghna Reddy¹, D.Vaishnavi², Salandri Shirisha³

Dr. M. Swapna⁴

Department of Computer Science and Engineering, Stanley College of Engineering and Technology for Women, Telangana, India

Abstract.

Dactylology is the technique of communicating with signs made with fingers, especially in the manual alphabets used by the deaf. Sign Language is one of the oldest and most natural form of language for interaction between normal and deaf & mute people, but since most people do not know sign language and interpreters are very difficult to find, we have come up with a real time method using neural networks for fingerspelling based American sign Language. Sign language is a visual language and consists of 3 major components: Fingerspelling, Word level vocabulary and Facial expressions.

In our project we basically focus on producing a model which can recognise Fingerspelling based hand gestures in order to form a complete word by combining each gesture.

We follow 4 steps in the hand gesture recognition: i) Data Acquisition, ii) Data preprocessing, iii) Feature extraction iv) Gesture classification. We aim to collect 600 RGB images each for 26 alphabets and apply a filter called Gaussian blur to each of the images. This filter gives a black & white outline of the hand gesture which makes it easier for computation. The images are then passed through convolution neural network classifier which predicts the class of the hand gestures. Our method provides 95% accuracy for the 26 letters of the alphabet

Keywords: Dactylology, Image Recognition, Gaussian Blur Filter, Image Processing, Deep Learning, Convolutional Neural networks.

1. Introduction

1.1 About Project

Sign language is a form of communication used by people with impaired hearing and speech. People use sign language gestures as a means of non-verbal communication to express their thoughts and emotions. Sign languages generally do not have any linguistic relation to the spoken languages of the lands in which they arise. The correlation between sign and spoken languages is complex and varies depending on the country more than the spoken language. We use a custom recorded American Sign Language data set based on an existing data set for training the model to recognise gestures. We propose to use a CNN (Convolutional Neural Networks) named Inception to extract spatial features from the video stream for Sign Language Recognition (SLR). American Sign Language (ASL) is a characteristic language that fills in as the prevalent sign language of Deaf people. American Sign Language (ASL) is a visual language. With marking, the brain processes phonetic data through the eyes. The shape, arrangement, and movement of the hands, just as facial

expressions and body movements, all play significant parts in passing on data, ASL has the same linguistic properties as spoken languages. We try understanding the precise meaning of deaf and dumb people's symbolic gestures and converting it into understandable language (Text) and finally into speech using Google Text to Speech (gTTS) module.

1.2 Objectives of the project

The aim is to develop a user-friendly human computer interface (HCI) where the computer understands the human sign language. There are various sign languages all over the world, namely American Sign Language (ASL), French Sign Language, etc and work has been done on other languages all around the world. But, we focus on converting American Sign Language into speech, as ASL is a global sign language and can be understood by people all over the world. In recent years there has been tremendous research done on hand gesture recognition.

With the help of literature survey done we realised the basic steps in hand gesture recognition are: - Data acquisition, Data preprocessing, Feature extraction, Gesture classification.

1.3 Scope of the project

We are planning to achieve higher accuracy even in case of complex backgrounds by trying out various background subtraction algorithms. We are also thinking of improving preprocessing to predict gestures in low light conditions with a higher accuracy.

2. Literature Survey

2.1 Existing System

Sign Language is the most natural and expressive way for the hearing impaired people. Many new techniques have been developed recently in this area. But the existing system uses Linear Discriminant Analysis (LDA) algorithm was used for gesture recognition and recognised gesture is converted into text. LDA mainly used in statistics, pattern recognition and machine learning. It is used to find a linear combination of features that characterises or separates two or more classes of objects or events.

The current system has collected around 10 images for each sign. These images are included in the training and testing database. The captured image at a distance is adjusted by the signer to get the required image clarity. Due to fewer images given for training, the accuracy of such a system drops to 85-88%. The image components are extracted by Morphological Filtering tools which are useful for representation and description of shape. The features extracted from the segmentation operation used for gesture recognition. The smooth contour is obtained by removing the noise from the images with Morphological filtering techniques.

2.2 Proposed System

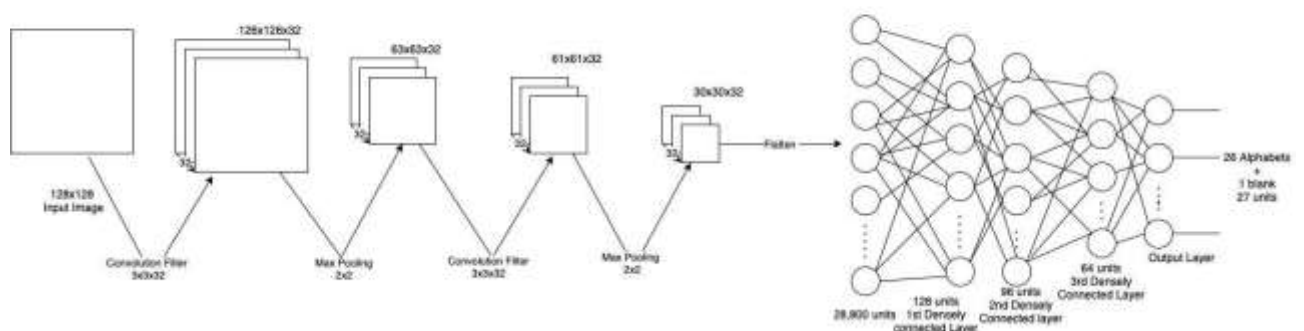
American Sign Language is a predominant sign language and is widely used by people all over the world. ASL alphabets are quite different from the gestures, alphabets involve manual letters from A-Z with each letter shown to convey a message, while gestures convey an entire emotional together. Creating an human interactive interface that takes ASL as the input and converts it into speech can be used by people all over the world without any barriers. The proposed system with a live video input is subjected to the trained sign recognition model, trained victimisation neural networks of deep learning, the model acknowledges the alphabet, combines recognised letters to words and finally combines words to sentences. A Hunspell spell checker is also included to suggest or correct words during the recognition process. The sentences are converted to speech/audio by using the gTTS library.

3. Proposed Architecture

The proposed architecture presents a model which captures images from the web camera, the image frame through which the desired input has to be taken can be manipulated from the software. The Image is then processed in the OpenCV platform to recognise the gestures through image processing algorithms. Spell checker package is included to correct the spellings in case of some erroneous inputs. The text is then passed to the gTTS library that converts the opened text into audio file.

The user gives input from the video capture with the help of the in-built camera on the computer. This camera helps in capturing the required images for each alphabet. This involves the image acquisition phase where the input from the user is taken. These images acquired are then preprocessed. The preprocessing involves converting the images into gaussian blur images.

Dactylogy Recognition system Training Architecture



1st Convolution Layer -Input 128*128 image, using 32 filters, output 126*126 image. 1st Pooling Layer - Downsampled using 2*2 max pooling to 63*63 pixels image. 2nd Convolution Layer - Input 63*63 image, using 32 filters, output 60*60 images. 2nd Pooling Layer - Downsampled using 2*2 max pooling to 30*30 images. The outcome of the CNN layers is 28,800 units. The 1st Fully Connected Dense layer consists of 128 units and this is followed by 2 fully connected layers of 96 units and 64 units respectively. Finally, there is the output layer which consists of 27 units, one for each alphabet and last one neuron for the blank space.

4. Implementation

4.1 Data Collection

We created a dataset which has 27 gestures and these gestures are made up of 26 alphabets and 1 blank space. There are 500 images for each alphabet in the 27 different folders. From the 13,000 images, 11070 images were given for training and 2755 images are given for testing.

4.2 Preprocessing

In pre-processing the image data is improved which reduces unwanted deviation or enhances image features for further processing. Preprocessing is also referred to as an attempt to capture the important pattern which expresses the uniqueness in data without unwanted data which includes cropping, resizing and grey scaling. Cropping removes the unwanted parts of an image to improve framing.



4.3 Designing the CNN (Convolution Neural Networks)

This step is the most important part of the entire process as we design the CNN through which we will pass our features to train the model and eventually test it using the test features. We have used a combination of several different functions to construct CNN which we will discuss one by one. Building and training a CNN in Keras:

Sequential() - A sequential model is just a linear stack of layers which is putting layers on top of each other as we progress from the input layer to the output layer.

- `classifier.add(Conv2D())` - This is a 2D Convolutional layer which performs the convolution operation as described at the beginning of this post. To quote Keras Documentation "This layer creates a convolution kernel that is convolved with the layer input to produce a tensor of outputs." Here we are using a 3x3 kernel size and Rectified Linear Unit (ReLU) as our activation function.

- `classifier.add(MaxPooling2D())` This function performs the pooling operation on the data_{SEP} as explained in Literature Review. We are taking a pooling window of 2x2 with 2x2 strides in this model.

- `classifier.add(Flatten())`- This just flattens the input from ND to ID and does not affect the batch size.

`classifier.add(Dense())`- According to Keras Documentation, Dense implements the operation: $\text{output} = \text{activation}(\text{dot}(\text{input}, \text{kernel}))$ where activation is the element-wise activation function passed as the activation argument, kernel is a weights matrix created by the layer. In simple words, it is the final nail in the coffin which uses the features learned using the layers and maps it to the label. During testing, this layer is responsible for creating the final label for the image being processed.

4.4 Algorithm

Our approach uses two layers of algorithms to predict the final symbol of the user. data_{SEP} Algorithm Layer 1: Apply Gaussian blur filter to the frame to get the processed image after feature extraction. This processed image is passed to the CNN model for and if a letter is detected for more than 50 frames then the letter is printed and considered for forming the word. Space between the words is taken as blank symbol.

Algorithm Layer 2: The symbols which look alike of each other, special classifiers will be applied to them depending on the accuracy obtained.

4.5 Tkinter Implementation

Tkinter is lightweight and relatively painless to use compared to other frameworks. This makes it a compelling choice for building GUI applications in Python, especially for applications where a modern sheen is unnecessary, and the top priority is to quickly build something that's functional and cross-platform. The foundational element of a Tkinter GUI is the window. Windows are the containers in which all other GUI elements live. These other GUI elements, such as text boxes, labels, and buttons, are known as widgets. Widgets are contained inside of windows.

In our implementation a video capture is opened which takes input from the user and displays the recognised letter in the form of character. This character is then passed to the word. On receiving blank as an input, the obtained word is passed to the sentence. On clicking the click to convert into speech button, the sentence is converted to speech.

5. Results

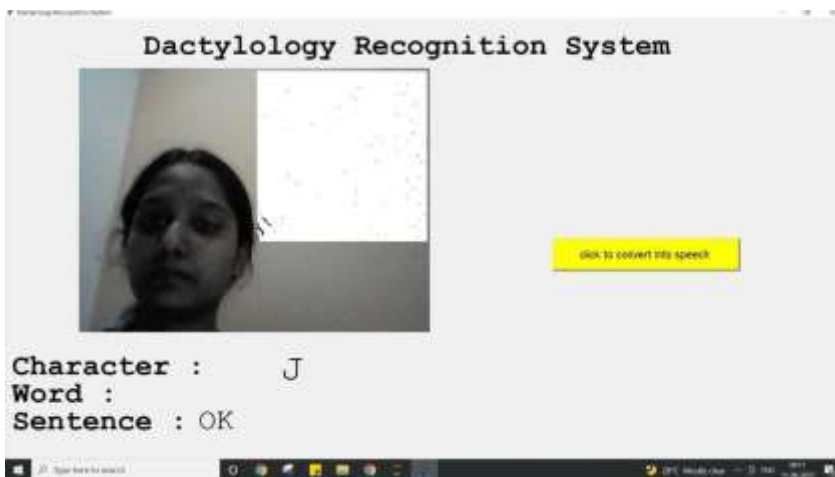


Fig 5.1: Output for the given Sign in the form of sentence

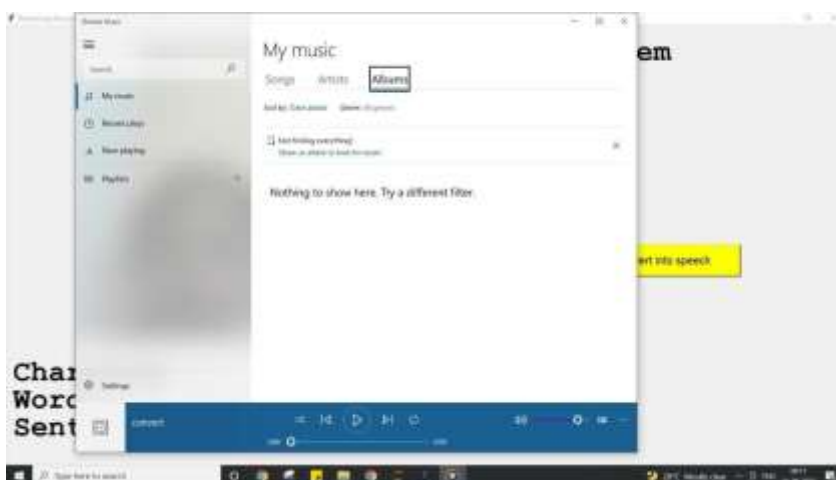


Fig 5.2: Sign Language to Speech Conversion

6. Conclusion

In this project, a functional real time vision based American sign language recognition for Deaf and Dumb people have been developed using ASL alphabets. We achieved final accuracy of 95% on our dataset. We are able to improve our prediction after implementing two layers of algorithms in which we verify and predict symbols which are more similar to each other.

We started from collecting the data by using OpenCV video capturing. This followed preprocessing the data and smoothing the images by using Gaussian Blur Filtering. The cleaned data was then trained using 2 CNN layers and 3 Densely connected layers. Training was done in 2 steps, one for all the alphabets and second step for alphabets looking similar to avoid confusion.

A front-end GUI was then developed using the Tkinter window to capture inputs. The output is a sentence which is converted to audio, on click of a button present on the window. This way we are able to detect almost all the symbols provided that they are shown properly and there is no noise in the background and lighting is adequate.

7. Future Scope

We are planning to achieve higher accuracy even in case of complex backgrounds by trying out various background subtraction algorithms.

We are also thinking of improving the preprocessing to predict text in low light conditions with a higher accuracy. Currently, our project uses manual alphabets where each letter is shown to combine it to a word and then into a sentence, we plan to add gestures in the future which would convey emotions like Thank you, I Love you, Hello etc. We plan to add dactylography recognition to online meetings as well using which deaf and dumb people and communicate, an algorithm can be applied to online meetings where a gesture shown is converted to text and shown to the others in the meeting.

Finally, we would like to improve the time efficiency of our system in order to make it appropriate to use in different applications.

8. References

- [1] T. Yang, Y. Xu, and "A. , Hidden Markov Model for Gesture Recognition", CMU-RI-TR-94 10, Robotics Institute, Carnegie Mellon Univ.,Pittsburgh,PA, May 1994.
- [2] Pujan Ziaie, Thomas Müller , Mary Ellen Foster , and Alois Knoll"A Naïve Bayes Munich,Dept. of Informatics VI, Robotics and Embedded Systems,Boltzmannstr. 3, DE-85748 Garching, Germany.
- [3]https://docs.opencv.org/2.4/doc/tutorials/imgproc/gaussian_median_blur_bilateral_filter/gaussian_median_blur_bilateral_filter.html

4. [4] Mohammed Waleed Kalous, Machine recognition of Auslan signs using PowerGloves: Towards large-lexicon recognition of sign language.
5. [5] aeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks-Part-2/
6. [6] <http://www-i6.informatik.rwth-aachen.de/~dreuw/database.php>
7. [7] Pigou L., Dieleman S., Kindermans P.J., Schrauwen B. (2015) Sign Language Recognition Using Convolutional Neural Networks. In: Agapito L., Bronstein M., Rother C. (eds) Computer Vision - ECCV 2014 Workshops. ECCV 2014. Lecture Notes in Computer Science, vol 8925. Springer, Cham.
8. Viswanatha Reddy, Dr. Elango NM and Dr. C Kishor Kumar Reddy, “ Internet of Things Based Early Detection of Diabetes Using machine Learning Algorithms:DPA”, International Journal of Innovative Technology and Exploring Engineering, 2019
9. Viswanatha Reddy, Dr. Elango NM and Dr. C Kishor Kumar Reddy, “Diabetes KaggleDataset Adequacy Scrutiny using Factor Exploration and Correlation”, International Journal of Recent Technology and Engineering, 2019
10. Viswanatha Reddy, Dr. Elango NM and Dr. C Kishor Kumar Reddy, “Diabetes Diagnosis and Prognosis using Machine Learning Approaches: A Survey, “ International Journal of Advanced Science and Technology, 2019
11. Kishor Kumar Reddy C and Vijaya Babu B, “ISPM-OC: Improved Snow Prediction Model Using Optimal k-Means Clustering and Decision Tree to Nowcast Snow/No-Snow”, International Journal of Control Theory and Applications , 2017
12. (http://serialsjournals.com/abstract/41550_14-c._kishor_kumar_reddy.pdf)
13. Kishor Kumar Reddy C and Vijaya Babu B, “ISLIQ-OC: Improved Supervised Learning in Quest using Optimal k-means Clustering Mechanism to Nowcast Snow/No-Snow”, International Journal of Control Theory and Applications , 2017.
14. (http://serialsjournals.com/abstract/61260_09-c._kishor_kumar_reddy.pdf)
15. R Gangadhar Reddy, M. Srinivasa Reddy, P R Anisha, Kishor Kumar Reddy C, “Identification OF EARTHQUAKES USING WAVELET TRANSFORM AND CLUSTERING METHODOLOGIES”, International Journal of Civil Engineering and Technology, 2017. (http://www.iaeme.com/MasterAdmin/UploadFolder/IJCIET_08_08_067/IJCIET_08_08_067.pdf)
16. B Subbarayudu, Srija Harshika D, E Amareswar, R Gangadhar Reddy, Kishor Kumar Reddy C, “Review and Comparison on Software Process Models”, International Journal of Mechanical Engineering and Technology, 2017.
17. (http://www.iaeme.com/MasterAdmin/UploadFolder/IJMET_08_08_105/IJMET_08_08_105.pdf)
18. Kishor Kumar Reddy C and Vijaya Babu B, “ISLGAS: Improved Supervised Learning in Quest using Gain Ratio as Attribute Selection Measure”, International Journal of Control Theory and Applications, 2016.
19. Kishor Kumar Reddy C and Vijaya Babu B “A Survey on Issues of Decision Tree and Non-Decision Tree Algorithms”, International Journal of Artificial Intelligence and Applications for Smart Devices, 2016.
20. Kishor Kumar Reddy C, Rupa C H and Vijaya Babu B, “ISLIQ: Improved Supervised Learning in Quest to Nowcast the presence of Snow/No-Snow”, WSEAS Transactions on Computers, 2016.
21. B Subbarayudu, Srija Harshika D, E Amareswar, R Gangadhar Reddy, Kishor Kumar Reddy C, “Review and Comparison on Software Process Models”, International Journal of Mechanical Engineering and Technology, 2017.
22. (http://www.iaeme.com/MasterAdmin/UploadFolder/IJMET_08_08_105/IJMET_08_08_105.pdf)
23. Kishor Kumar Reddy C and Vijaya Babu B, “ISLGAS: Improved Supervised Learning in Quest using Gain Ratio as Attribute Selection Measure”, International Journal of Control Theory and Applications, 2016.



24. Kishor Kumar Reddy C and Vijaya Babu B “A Survey on Issues of Decision Tree and Non-Decision Tree Algorithms”, International Journal of Artificial Intelligence and Applications for Smart Devices, 2016.
25. Kishor Kumar Reddy C, Rupa C H and Vijaya Babu B, “ISLIQ: Improved Supervised Learning in Quest to Nowcast the presence of Snow/No-Snow”, WSEAS Transactions on Computers, 2016.
26. (<http://www.wseas.org/multimedia/journals/computers/2016/a055705-872.pdf>)
27. Kishor Kumar Reddy C and Vijaya Babu B, “ISPM: Improved Snow Prediction Model to Nowcast the Presence of Snow/No-Snow”, International Review on Computers and Software, 2015.
28. (<http://www.praiseworthyprize.org/jsm/index.php?journal=irecos&page=article&op=view&path%5B%5D=17055>)