An Approach To Translate Hand Gestures Into Telugu Text

J SreeDevi, Dr M Rama Bai, Mohammed Maqsood Ahmed

Abstract: Communication an indispensable part of human activities which is expressed both verbally and non-verbally is an ease for many of us but there are many like deaf and dumb for whom it is a challenge. It is a hurdle which is hampering their talent and decreasing their productivity (proficiency); Engagement with such specially abled persons can only happen if we know the sign language which is an organized form of communication or if there is an interpreter in between; This study attempts to decrease these hurdles to some extent by building a Gesture Recognition Model through Convolutional Neural Networks(CNN) which can predict the static hand gestures made by them and convert them into Telugu text; which will be handy in of Telugu speaking states. People who don't understand sign language can easily understand what they have to say and therefore there won't be any need of Interpreter. This study follows the American Sign Language which is the most widely used language across the globe wherein each alphabet/word is assigned a sign to communicate.

Index Terms: Gesture Recognition, CNN, Sign Language, Telugu Text

1. INTRODUCTION

As any other language, Sign language comes with its own rules and conventions in this communication is made through designated form of hand gesture and body postures; According to 2011 census of India there are about 1.3 million people who suffer from “hearing impairment” and there are 1.8 million people who are “dumb” according to India’s Association for Dumb,. For all such people sign language is the first language and their mother language while there are many sign language being used across the world, of whom some have obtained legal sanction, among them American sign language prevails over others because of its easiness and agility. The goal of this study is to build Convolutional Neural Network model which is able to classify which letter, gesture of the American Sign Language (ASL) alphabet is being signed, given an image of a signing hand through webcam and convert the predicted letter/gesture into Telugu text. This study is a first step towards building a possible sign language translator, which can take communications in sign language and translate them into written and oral language. Such a translator would greatly lower the barrier for many deaf and mute individuals to be able to better communicate with others in day to day interactions. This goal is further motivated by the isolation that is felt within the deaf community. Loneliness and depression exist in higher rates among the deaf population, especially when they are immersed in a hearing world. Large barriers that profoundly affect life quality stem from the communication disconnect between the deaf and the hearing. Some examples are information deprivation, limitation of social connections, and difficulty integrating in society. Most research implementations for this task have used depth maps generated by depth camera and high-resolution images. The objective of this paper was to see if neural networks are able to classify signed ASL letters using simple images of hands taken with a personal device such as a laptop webcam. This is in alignment with the motivation would make a future implementation of a real time ASL-to-oral/written language translator practical in an everyday situation.

Developing a model involving computer vision always presents some challenges such as:
- Sensitivity of the background in image.
- Occlusion (Half jetted images)
- Boundary detection

The proposed study attempts to evince the sign language into a model, which recognizes the gestures effectively and which converts the predicted gesture into Telugu text by tapping the power of deep learning using Convolution Neural Networks; The advancement in the field of computer vision coupled with deep learning acts as a paragon for this problem. The problem have been in the hands of various researchers in the past and to sum up them we can classify their work majorly under two domains:
Sensor Based: The data which used for classification is collected through different sensors; the most revered solution under this domain is Data gloves or LED’s that came up in 1977 however the use of external device hinder the natural movement and solution is not economically feasible.
Vision Based: The advent of machine learning, artificial intelligence paved the path for this approach; the central idea is to capture the image and process it over filters and in order to recognize, The use of machine learning involves developing your own filters or use the established filters and then deploy them to linear classifiers or other algorithms but if we proceed with Deep learning the problem of external filters drops out. The rise of research in convolution neural networks has set the benchmark for gesture recognition techniques; We build a sign language recognition system using CNN through which one can classify the gestures/letter efficiently and then convert the text into Telugu language. A CNN model will be created which will be trained on the Kaggle Sign Language dataset along with user-made hand gestures dataset; the model will be assessed by its accuracy on the test-dataset and then model will be utilized to predict the unknown gestures recorded through the webcam. The predicted gesture by the model will be given as input to the Google translation object which will convert the gesture text into Telugu format.

J Sreedevi, Assistant Professor, CSE, MGIT, Telangana, India.
E-mail: jsreedevi_cse@mgit.ac.in
Dr M Rama Bai, Professor, CSE, MGIT, Telangana, India.
E-mail: rama@mgit.ac.in
Mohammed Maqsood Ahmed, Student, CSE, MGIT, Telangana, India.
E-mail: maqsoodhuman@gmail.com
2. SYSTEM DESIGN

The system basically comprises of the following steps:

(i). Creating a model which classifies different gestures; The model will be trained by a predefined data set (Kaggle sign language gestures data set + own gestures) containing different gestures after pre-processing.

(ii). Images of the gesture is captured using a simple web camera and they are stored in test folder.

(iii). The acquired images (test folder) are then used as input to the gesture classification model created in step (i).

(iv). The model will classify the image/gesture into one of the predetermined gestures.

(v). The predicted gesture will be converted into Telugu Text

The flow of the system is shown in the Figure 1 below.

The study is structured into 4 distinct functional modules namely:

- Data Processing
- Training
- Classify Gesture
- Converting the Predicted gesture text into Telugu Text

Data Processing

This part constitutes functions to load the Raw Image Data and save the image data as numpy arrays into file storage. The NumPy array file will be send further to preprocess the image by resizing/rescaling the image and applying filters. During training the processed image data was split into training, validation, and testing data.

Data Collection

The primary source of data for this work was the compiled dataset of American Sign Language (ASL) called the ASL Alphabet obtained from Kaggle along with the dataset created by us. In this we have only used gestures from ‘A’ to ‘M’ of ASL dataset and along with that we have used created gestures namely “Win”, “I love you”, “Best of Luck”, “Hello”, “Protect”. In short, our dataset is comprised of 1020 images which are of the size 320x240. There are 17 total classes, each gesture with 60 images. These photos as show in figure 2 were then cropped, rescaled, and labeled for use.

![Figure 2: Gestures present in the dataset](image)

The above Table 1. represent the order of attributes present in .csv file. The attributes are further used to crop the image and extract gesture from it. For collecting the .csv file we have utilized the method genfromtext() available in numpu library of python; which allows us to extract the text data. The main idea is to create a function that can crop a unique image; and then apply it to crop all the images extracted.

We have assumed that each user has exactly same number of images and each gesture have same number of gestures in it. Therefore:

<table>
<thead>
<tr>
<th>image</th>
<th>topleft x</th>
<th>topleft y</th>
<th>bottom right x</th>
<th>bottom right y</th>
</tr>
</thead>
<tbody>
<tr>
<td>user_3/A0.jpg</td>
<td>124</td>
<td>18</td>
<td>214</td>
<td>108</td>
</tr>
<tr>
<td>user_3/A1.jpg</td>
<td>124</td>
<td>18</td>
<td>214</td>
<td>108</td>
</tr>
<tr>
<td>user_3/A2.jpg</td>
<td>123</td>
<td>19</td>
<td>213</td>
<td>109</td>
</tr>
<tr>
<td>user_3/A3.jpg</td>
<td>122</td>
<td>21</td>
<td>212</td>
<td>111</td>
</tr>
<tr>
<td>user_3/A4.jpg</td>
<td>122</td>
<td>20</td>
<td>212</td>
<td>110</td>
</tr>
<tr>
<td>user_3/A5.jpg</td>
<td>96</td>
<td>35</td>
<td>176</td>
<td>115</td>
</tr>
<tr>
<td>user_3/A6.jpg</td>
<td>93</td>
<td>39</td>
<td>173</td>
<td>119</td>
</tr>
</tbody>
</table>
total images = number_of_users * images_per_user

Traversing over the range of total_images we process each image of corresponding user by giving it as input to the crop_function which will return the cropped image; At the end we will have an array of cropped images.

The crop_function utilizes the attributes of the csv file to crop the image and after cropping image we resize the image of the order 128*128*3

Using one-hot encoding for splitting data to test and train

As machine learning algorithms doesn’t understand categorical data we need to mould our data in a manner which is easily understandable my machines and one such representation is one hot encoding; it represents categorical data as binary vectors wherein each values index will be represents as 1 while others are 0.

The output of the one hot encoding is in the following manner. [[[1. 0. 0. … 0. 0. 0.] [1. 0. 0. … 0. 0. 0.] [1. 0. 0. … 0. 0. 0.] … [0. 0. 0. … 0. 0. 1.] [0. 0. 0. … 0. 0. 1.] [0. 0. 0. … 0. 0. 1.]]

Splitting on Dataset

Splitting of the dataset is one of the precursor required to work on machine learning, neural network models it separates the cases into two or more new datasets; Following the convention we split our data into tow forms one for training and other for testing.

The splitting of dataset has the following parameters

Training data corresponds to data which will be utilized in training the module; it has two parts i.e. input data and the value of the input data.

• x_train corresponds to the input data for training.
• y_train corresponds to the values of the given input data.

Following the same manner our testing data set corresponds to test input and test

• x_test is the input data for testing while.
• y_test corresponds to label/values of the given test inputs.

We have split the dataset in such a way that up to penultimate: user data will be used for training except the last one which will be you used for testing

In order to attune our data in a manner accepted by our model we need to consider tow thigs i.e. no of images for each label and size of image (width, height, num_of_pixels) should be constant.

CNN Model

Here we have utilized keras over tensor flow to create our CNN model with a backend of tensor flow which at a high level, allows us to work on graph data flows. Convolutional neural network (CNN) present to us a modest way to solve this problem; it comprises of input, hidden and output layers with nuances in each layer. Hidden layer gives us the opportunity to add multiple convolution layers, pooling layers, fully connected layers and normalization layers.

Architecture of the Model

The model used in this classification task is a fairly a basic implementation of a Convolutional Neural Network (CNN). This study consists of convolution blocks containing two 2D Convolution Layers starting with Batch normalization, using ReLU as our activation function, followed by Max Pooling. These convolution blocks are repeated 4 times and followed by Fully Connected layers that eventually classify into the required categories. The kernel sizes are maintained at 5 X 5 throughout the model. Training Deep Neural Networks is an encumbrance task given the facts that uniformity of the input and its parameters differ from each other in layers because of training problems such as diminishing learning rate kicks in and initialization of parameters makes it a hard tool to train the model; To co-opt this situation, we have utilized Batch Normalization; which helps us in making uniformity among the inputs over a network; it can be applied prior or post activation. It expeditiously helps in training and reducing the error. The activation function which determines whether a neuron should be activated or not by adding its weighted sum with some bias; in this study we have utilized ReLu activation function to do this operation; ReLU is used for activation except the last layer; though it is a non-liner function it appears to work as linear, making it easier to manage and work. Anything negative sumps up to 0.

Figure 3: Model Architecture

At the end of the model we have utilized SoftMax activation function which is one of the logistic regression functions who maps the input values into vector series that is projected over a probability distribution and whose sum of the values will be equal to one.
Pooling which is responsible for reducing the size of the input. We have make use of max pooling of size (2,2) which does striding over the image and takes the max value of the each stride as the output therefore at the end we get a image which is reduced in size; it can be visualized through figure 6.

Conversion of Text
The predicted gesture text is processed further by giving it to googletrans library we convert the text into telugu language and represent it using the one of the suitable font present in our system. After classifying the gesture into any one of the classes we have utilized the googletrans library in python to covert the gesture/letter in Telugu format. To convert the text first we first create a translator object which utilizes the method Translator.translate(); the destination for translation is given as ‘te’ which corresponds to Telugu language.

The database which we have created has been stored in format of characters while there meaning is stored in a dictionary
The predicted class is replaced with its key-value pair or literal translation of character is done.
The font utilized to print is NIRMALA.TTF which seems to be standard Telugu font across the windows systems.

3. TRAINING AND VALIDATION
The model is trained with hyper parameters such as learning rate, batch size, image filtering, and number of epochs. The configuration used to train the model is saved along with the model architecture for future evaluation and tweaking for improved results. Within the training loop, the training and validation datasets are loaded as Data loaders and the model is trained using rmsprop with categorical_crossentropy Loss. The model is evaluated every epoch. Upon finishing training, the training and validation error and loss is saved to the disk.

a. Training
Our model is trained using RMSProp (Root mean square propagation) optimizer and categorical_crossentropy. RMSPropOptimizer is known to accelerate the learning process by penalizing the update of those neural network parameters that make the estimate of the cost function oscillate too much; Learning rates gets modified by default and it opt for a other learning rate for each parameter. A loss function maps decisions to their associated costs and in this model, we have utilized Categorical cross entropy which maps label into one category. In other words, an example can belong to one class only.

b. Validation and Accuracy
Keras provides. fit() method paves the way to train our own deep learning models; The call to .fit is makes two primary assumptions
1) Our entire training set can fit into RAM
2) There is no data augmentation going on (i.e., there is no need for Keras generators)
The arguments of the fit method include batch size which indicates to and from motion of the entire dataset along the network since our data set is too large to be forwarded at one we categorize the dataset into batches. The other arguments include training data, set and validation data.

During the training we observe as the no of epochs increase the training accuracy and validation increases simultaneously while on the other hand the loss and val_loss decreases as its
proceeds this indicates that our model is neither under-fitted nor over-fit.

We have used different test cases to evaluate and determine the accuracy of our model and we have achieved a high accuracy of 93% and on an average we have been getting an accuracy between 85-94 percent.

![Figure 8: Model Accuracy](image)

![Figure 9: Model Loss](image)

Confusion Matrix:

![Figure 10: Confusion Matrix](image)

Classification Report:

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed</td>
<td>94%</td>
</tr>
<tr>
<td>[1](Microsoft Kinect)</td>
<td>91.70%</td>
</tr>
<tr>
<td>[2](SVM and K-nearest neighbours)</td>
<td>63.20%</td>
</tr>
<tr>
<td>[3](Karhunen-Loeve Transform)</td>
<td>95%</td>
</tr>
<tr>
<td>[4](Dynamic Bayesian network)</td>
<td>99%</td>
</tr>
</tbody>
</table>

Individual accuracy:

![Figure 12: Accuracy of each class](image)

c. Testing

To determine whether our preprocessing of images actually results in a more robust model, we have verified on a test set comprised of images from the original dataset and we have also verified on the collected image data from the webcam in...
4. CONCLUSION

Sing Language Recognition is not a new problem but the advent of CNN has shifted the goalpost of this problem to a new level; In our study we have substantially shown the CNN is the most expeditious way to handle Gesture Recognition Problems; Our study includes gestures from different users which indicates a composite dataset utilized but still there is a room for improvement in gestures collection from different backgrounds thorough which we can enhance the model more.

**Future Scope**
- Model can be enhanced more by incorporating various nuances such as more diverse gestures from different users in different background.
- Deployment of the model on a web application can unleash the true potential use of our study.
- A two-way communication model which can be used for communication directly.

**REFERENCES**