IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 2, April 2022
Impact Factor: 6.252

Sign Language Translator

Purva Vijay Bhave[1], Charusheela Rajendra Deshmukh[2], Ulka M. Shirole[3]
Students, Department of Information Technology[1,2]
Professor, Department of Information Technology[3]
A.C. Patil College of Engineering, Mumbai, Maharashtra, India

Abstract: *Communication is an integral part of today's world but still it is not convenient for everyone i.e. communication for deaf and dumb people has always been troublesome. This paper focuses on creating a sign language translator which converts text or voice input into sign language video feed and also provides a real-time sign language detection, thereby creating a two-way form of communication. This paper discusses use of machine learning and deep learning for real time detections. The main purpose of this translator is to eliminate communication barriers that deaf and dumb people face by providing them a tool to communicate in social as well as corporate world effectively and efficiently.*

Keywords: Sign language translator, Neural Network, Computer vision, Deep learning, Transfer learning, Mediapipe, LSTM.

I. INTRODUCTION

Sign language is a language used by deaf and dumb people to communicate. Sign language makes use of hand gestures and facial expressions to convey the messages. Though it is helpful for people with hearing disabilities, sign language is not something that is generally taught in schools so not everyone knows how to use it, which gives rise to communication barrier among deaf and dumb people and others. This is not only limited to everyday communication but it also affects social, educational and corporate lives of people with hearing disabilities [4]. To deal with this problem, this paper presents an application, rather a tool that people can use to translate sign language. This application will be bidirectional i.e. it will be able to take in text or voice input and translate it into a sign language video feed, along with that it can make real-time detections of signs thereby providing an effective way for people to communicate. Leveraging the power of machine learning and computer vision, these detections are much faster and efficient so as to not affect the conversation among individuals and thus eliminating lag as much as possible. This application will not only translate alphabets but also digits and actual words which have signs associated with them [4].

II. LITERATURE SURVEY

A. Sign Language Semantic Translation System using Ontology and Deep Learning

This paper focuses on enhancement of sign language detection by introducing ontology along with deep learning. Arabic language dataset involving alphabets and some words is used for training [1]. CNN, a convolutional neural network with 4 hidden layers is used to train the data. Before feeding data into the model, steps such as pre-processing of image data is necessary to make sure that input size of each image stays same and the image data is in suitable format for underlying network. Once pre-processing is done, training can be started. Since dataset was small for a CNN to be trained on, the dataset is augmented. This is a process where the training data is expanded by either shifting few pixels, rotating the image, zooming in or out, etc. Once a single image is obtained from multiple perspectives the model is trained on the original and augmented images collectively. The accuracy obtained using this method was about 98.06%. The only drawback of this implementation is that it uses static detection but most sign in sign language need movement which won't be detected using this methodology [1].

B. Neural Sign Language Translation by Learning Tokenization

This paper introduces core mechanisms and requirements for building an efficient and accurate sign language translating system. It proposes use of Neural Machine Translation and Tokenization [2]. All translation tasks, even English to some other language rely highly on context of the statement. Based on context, accurate translations can be made. The dataset used is generated using keypoints provided by OpenPose which is an opensource software that can extract body and hand

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 2, April 2022
Impact Factor: 6.252

keypoints which are then used to train the model. Using OpenPose the data is collected and then sent into the model. Large dataset is created to make sure that the learned weights and hence the model is robust and accurate [2].

The model used for training is LSTM, Long Short-term memory which is great for context-based detections like translations. Along with this method an alternative of using tokenization is proposed. Tokenization is a method wherein each word is assigned a weight vector and these weights are used to create an embedding which is then used to train the model. Tokenization is mostly used for language related task like sarcasm detection or sentiment detection from a text input and other similar tasks [2].

C. Sign Language Recognition with Transformer Networks

This paper introduces a different combination of feature extraction by using various methods like OpenPose which jointly detects presence of human body by using real-time multi-person system, hand keypoints, facial keypoints, foot keypoints on a single image and end-to-end learning with CNN, Convolution Neural Network to predict the correct label for gesture [3].This paper presents four different architectures used for Sign Language Recognition out of which three are based on transformer networks where at first Pose LSTM network is used with OpenPose features as a baseline which evaluates the impact of multi- head attention on accuracy of the neural network, the second method used is Pose Transformer Network which uses keypoints extracted by the OpenPose as an input for a multi- head attention for a classifier, third method used is Video Transformer Network, the feature extractor used here is a frame wise, 2D CNN which transforms each frame into a 512- dimensional vector. The fourth network architecture is based on LSTMs which uses end-to-end deep learning to extract features from video data and classifies signs based on these features. Initial results of this work indicates that the network trained using feature extractor by OpenPose estimation technique is able to significantly outperform the previous work [3].

III. METHODOLOGY

Standard machine learning tasks have same structure to approach a particular problem which involves data collection, data processing, scaling, normalization, splitting training and testing set (make sure sampling bias is avoided), model definition, training, and then finally evaluating model using test set and appropriate metrics.

3.1 Data Collection

The very first and crucial step is to get relevant data. Data collection is the process of gathering and measuring information on targeted variables in an established system, which then enables one to answer relevant questions and evaluate outcomes.

3.2 Data Cleaning and Pre-processing

Irrelevant and noisy data are either removed or processed [8]. For this project image data is used which is then converted into arrays so it is important to ensure that data used is not biased or the model won't generalize well and end up giving importance to irrelevant features. The data should be transformed in relevant format for the underlying network.

3.3 Normalizing Data

Normalizing data is an essential step as the goal of normalization is to change the values of numeric columns in the dataset to use a common scale, without distorting differences in the range of values or losing information.

3.4 Spliting Data

The data is split into training and testing set, training set is used to train the model while the testing set is used to evaluate model's performance. Splitting data early on into training and testing set makes sure that data does not leak into test state.

3.5 Model Selection

LSTM is selected generally for tasks where sequence or context of prediction is important which is the very first requirement of sign language translation, as changing order of words would either change its meaning or make it unreadable. Based on task at hand, regression or classification, multiple models are selected. CNN can be used [7] instead of LSTM wherein just static images are to be detected instead of signs where action is involved.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 2, April 2022
Impact Factor: 6.252

3.6 Training and Hyperparameter Tuning

A hyperparameter is a parameter whose value is used to control the learning process. Hyperparameter optimization or tuning is the problem of choosing a set of optimal hyperparameters for a learning algorithm that minimizes the cost function. While training, various parameters like loss function, activation function, learning rate, etc. are to be specified based on task at hand.

3.7 Evaluation

The testing set is used to evaluate the model to see model's performance on previously unseen data. Model's performance decides further steps based on whether it has overfit or underfit the data.

IV. IMPLEMENTATION

Standard machine learning tasks have same structure to approach a particular problem which involves data collection, data processing, scaling, normalization, splitting training and testing set (make sure sampling bias is avoided), model definition, training, and then finally evaluating model using test set and appropriate metrics.

4.1 Text/Voice to Sign Language

A. Collect Image Data and Store It

Various words used in day-to-day conversation are captured in sign language. Common words such as hello, thank you, please, etc. are the word for which images are captured. The images are captured using python's opencv module which is a module used for computer-vision tasks. A specific file structure needs to be followed for easy access while creating the video feed.

B. Input Processing

The input comes either in form of text or voice. The text input can be used as it is by splitting the words at spaces and then translating them individually but same cannot be done for voice input. The voice input is captured using speech recognition module which provides us a way to capture microphone input. The voice input is then converted to text which is then further processed in same way the text was processed. Precautions are taken to make sure that the voice and text input by user is accurate.

C. Script

Once input is in appropriate format i.e. a list of words, these words are then used as an index to search for relevant translation images, which are then stitched together to form a video feed of translation of the entire sentence. These images are stitched together to form a video using VideoWriter method provided by opencv.

D. Output

The video is then played based on user's command. The video can also be replayed or closed. When new input is provided the old video gets overwritten to avoid unnecessary wastage of space. These videos can also be played at user's desired speed.

4.2 Sign Language to Text/Voice

A. Requirements and Dependencies

Mediapipe provides a pre-built hand, pose, face and many more detection package which are used to locate the keypoints. Tensorflow is also used for defining and training our LSTM model.

B. Extract and Store Keypoints

The mediapipe package is used to detect presence of hands and its keypoints. Mediapipe provides 21 keypoints for each hand. Once video feed begins, mediapipe detects keypoints which we then store in separate numpy arrays. Two arrays are required for each hand. Mediapipe's detection returns a dictionary of landmarks for each keypoint which consists of x, y and z co-ordinates. These co-ordinates are used to locate keypoint position for a specific sign and thus instead of just an

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 2, April 2022
Impact Factor: 6.252

image, complete action for a sign is detected. These keypoints are stored as numpy arrays which are then flattened as they are the input to our model. Along with extracting data, labels are generated as our network is supervised learning model. The dataset used is created by us. It has around 100 words so far where each word has 10 videos each with 20 frames wherein each frame has around 126 points as each hand has 21 keypoints with 3 co-ordinates. All in all for 100 words we have 1000 instances to train on.

C. Split the Data

Before any pre-processing some part of the data is kept separate to evaluate our model. Generally, about 20% of dataset is reserved for testing set but this percentage may vary based on the size of the dataset. Splitting the data helps determine how our model will act once launched. This gives us a basic idea about how our model will react when new instances are introduced.

D. Training the Model - LSTM

Training a model is a process where a specific mathematical function determines weights to be set to connections of the neurons which will provide accurate or close to target value predictions. In tasks where context or past data is important for predicting/detecting, LSTM, Long Short-term memory are used. The basic principle of LSTM is that they are type of recurrent neural network which processes data as it propagates forward as shown in "Fig. 1". For instance, if series of neurons are considered, each neuron depends on its previous neuron for output, i.e., some data or information from the very first neuron propagates to end thereby making it suitable to be used in sequence prediction problems such as translation, prediction of next word or speech recognition. Once model is defined, we compile it and then fit it on our training set for specific number of epochs. Each layer has an activation function such as ReLU, adam, etc. After epochs are completed, the model is then tested and evaluated. Callback is also described to facilitate early stopping.

E. Testing and Evaluation

Once model is trained for couple of epochs, it is now time to test accuracy of model. This is done by predicting values using test set as new instances. Since this data is unseen and unprocessed like real world data, accuracy obtained here is considered as main metric instead of the accuracy obtained while training. If accuracy while training is too large as compared to testing accuracy that means the model has overfit the data. Accuracy is not the only metric of evaluation, based on task at hand other metrics like confusion matrix, precision, recall, etc. can also be used, as for this model, categorical accuracy is used.

F. Real-Time Detection

To make real-time detection a separate function which launches webcam is created. This webcam then extracts each frame from the live feed and sends these frames in for detection by model. Once model detects signs in these frames it figures out the text or entire sentence based on probability of detection and prints it onto the screen. This text is also then processed to get rid of any grammatical errors.

G. GUI

Finally, integrating both components, namely text/voice to sign and sign to text/voice into a proper format is an essential step. Thus, a simple GUI is created wherein user can select their necessary translation options such as webcam or microphone which will thereby generate respective detection or translation of signs.

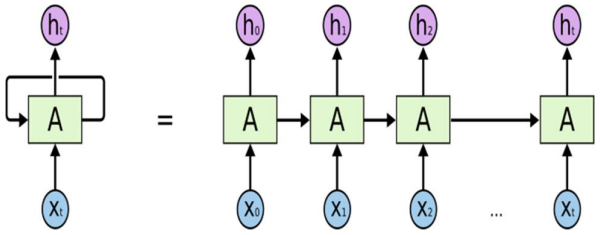

Figure: LSTM Network

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 2, April 2022
Impact Factor: 6.252

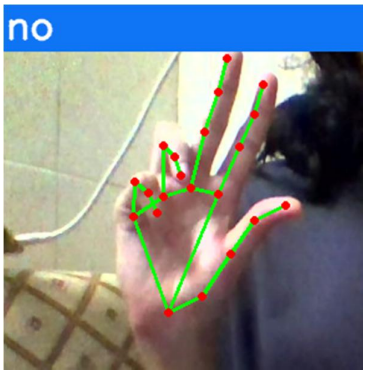

Figure: Sign Detection – "NO"

V. RESULT

5.1 Sign to Text/Voice Result

A. Dataset

Current dataset has 100 words where each word has 10 videos worth data and each video has 20 frames which has 126 points each. The current model is trained for 200 epochs but a callback function is provided in order to avoid overfitting and facilitate early stopping.

B. Model Summary

Three LSTM layers having 64, 128 and 64 neurons each are placed sequentially, followed by two dense layers with 64 and 32 neurons each. Lastly, a dense layer which will be the output layer has neurons corresponding to words we have trained on, in this case 100. All layers except for output layer has relu (Rectified Linear Unit) as an activation function, the output layer has softmax as its activation function. The model has adam as its optimizer and categorical crossentropy as its loss function..

C. Accuracy

So far about 100 words are trained using LSTM network with detection accuracy of about 92%. As dataset is further increased, accuracy will increase too, specifically categorical accuracy. The accuracy can be further increased as mentioned in future scope.

D. Sample

"Fig 2." Shows the sign detection. The model takes in every frame as an input and then detects respective sign based on parameters it learned while training. In "Fig 2" the keypoints from the frame are sent into trained model which then determines whether keypoints captured in real-time match the trained sign for– "NO".

E. Future Scope

Further improvements can be made by adding more layers or neurons, changing optimizer, generating more dataset, adding dropout layers, HLTSM[5], CNN along with ST-LSTM[6],etc. End goal is to train the model with optimum amount of words so as to facilitate basic conversation in day-to-day life.

VI. CONCLUSION

Our aim was thus to create a way to improve communication which is fundamental part of information technology and also to provide a significant contribution for more research in sign language detection. The solutions we have provided are structured and efficient. The surveys and research papers referred provide insights into what other techniques can be implemented to further better our solutions. The end goal is to create a system that can be utilized as a sign language translator in real-world. The system will have simple and easy to use GUI which makes it user-friendly and thus widens area of application and users. Having this system in real-world will definitely make a difference in lives of people with hearing disabilities. This application once deployed will widen its reach without any intervention on sole feature of its

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 2, April 2022
Impact Factor: 6.252

efficiency. This application will not only provide a new form of communication but also pave a way for more research in sign language translation.

ACKNOWLEDGMENT

For support and time-to-time guidance when needed we thank Prof. Ulka M. Shirole, Department of Information Technology.

REFERENCES

[1]. Elsayed, Eman K., and Doaa R. Fathy. "Sign language semantic translation system using ontology and deep learning." International Journal of Advanced Computer Science and Applications 11 (2020).

[2]. Orbay, Alptekin, and Lale Akarun. "Neural sign language translation by learning tokenization." arXiv preprint arXiv:2002.00479 (2020).

[3]. De Coster, Mathieu, Mieke Van Herreweghe, and Joni Dambre. "Sign language recognition with transformer networks." 12th International Conference on Language Resources and Evaluation. 2020..

[4]. Camgoz, Necati Cihan, et al. "Sign language transformers: Joint endto-end sign language recognition and translation." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2020.

[5]. (Guo, D., Zhou, W., Li, H., & Wang, M. (2018). Hierarchical LSTM for Sign Language Translation. Proceedings of the AAAI Conference on Artificial Intelligence, 32(1). Retrieved from https://ojs.aaai.org/index.php/AAAI/article/view/12235.

[6]. Q. Xiao, X. Chang, X. Zhang and X. Liu, "Multi-Information Spatial– Temporal LSTM Fusion Continuous Sign Language Neural Machine Translation," in IEEE Access, vol. 8, pp. 216718-216728, 2020, doi:10.1109/ACCESS.2020.3039539

[7]. Ankit Ojha, Ayush Pandey, Shubham Maurya, Abhishek Thakur, Dr. Dayananda P, 2020, Sign Language to Text and Speech Translation in Real Time Using Convolutional Neural Network, International Journal Of Engineering Research & Technology (IJERT) NCAIT – 2020 (Volume 8 – Issue 15).

[8]. System Theory (SSST), 2011 IEEE 43rd Southeastern Symposium on Priyanka MekalaYing GaoJeffrey FanA. DavariA. Davari.