Proceedings of 12th IOE Graduate Conference
Peer Reviewed
ISSN: 2350-8914 (Online), 2350-8906 (Print)
Year: 2022 Month: October Volume: 12

Word Level Nepali Sign Language Recognition Using Transformer

Shristi Heuju [a], Birodh Rijal [b]

[a] *Department of Electronics and Computer Engineering, Thapathali Campus, IOE, Tribhuvan University, Nepal*
[b] *Everest Engineering College, Pokhara University, Nepal*
[a] shristiheuju@gmail.com, [b] birodh.rijal@gmail.com

Abstract
With the seemingly distinct communication gap between the hearing impaired community and all the other remaining population, introducing sign language for the communication has definitely tear down this gap.The work concentrates on creating a vision-based system that will help identify the sign language hand gestures from the video sequences and help translate that into a readable format. The work focuses on two different methods. One work uses one of the architectures of convolutional neural network that shall train a classifier to classify spatial features of Nepali Sign Language words gesture. And, RNN approach shall be used for the classification of the temporal features of the word gesture. Another method utilizes the concept of transformer in combination with feature extraction approach to recognize the hand gestures made by the individual. The word level sign gestures captured to create a dataset is fed for train and test using DenseNet-121 architecture in combination with a RNN architecture of LSTM. Same extracted frames are then fed into DenseNet-121 architecture for feature extraction and finally into transformer for the sequential features recognition. The results obtained from use of combination of CNN and RNN architectures are compared and analyzed with the second approach to determine the best accuracy provided to recognize the sign gesture.

Keywords
Convolutional Neural Network, Deep Learning, Sign Language Gesture, Transformer

1. Introduction

Sign Language that is based on the gestures mad by movement of the body parts is used by the hearing impaired people as a means of communication. Similar to the spoken language, with different communities, ethnic groups, regional variation and linguistic nature these sign languages differ vastly and there exists no single universal sign language.

With the advancement in the machine learning techniques, machine based sign language recognition has become a hot topic as it benefits the community as well. This research targets to use deep learning based approach to recognize sign language gestures by feeding sign videos into the model.

2. Literature Review

The review was divided into three sections. The dataset's availability was investigated, and a few publicly available standard datasets that can be utilized to train and test the machine learning model were discovered. For the present methodologies and implementations utilized for sign language recognition, the existing scenarios of the study completed and recent trends were reviewed. In addition, there is literature on sign language recognition using various machine learning models, which can be enhanced and used further.

With the existing dataset on word level limited to very few number of words, sometimes it is impossible to apply in practice. To prepare WLASL dataset [1] the data collection is done from different sources with different data collection process. The dataset is constructed in a large-scale which is signer independent where the source of collection is internet. With this the meta information of temporal bounding box, signer annotations and sign dialect annotations are also included for the dataset preparation. For each video, the annotations was obtained where the video length varied from 0.36 seconds to 8.12 seconds with average length of overall videos being 2.41 seconds. 0.85 seconds of intra-class average standard deviation was obtained. The glosses were sorted in descending

fashion of sample of gloss. The dataset fashioned 2000 different ASL word with the help of over 100 participants.

Another dataset was created for Brazilian Sign Language [2] which was focused on the idea how the creation of dataset can be made cheaper for the image recognition process. The work focused on the creation of 2 dataset i.e., one with 30 fps which was splitted into 80:20 basis for train and test purpose. The words that is represented same in both singular and plural form was specified in one folder. The words that had no explicit meaning were included in ”formation” folder. During the creation of the dataset, each sign was made to repeat 2 times. The sign gestures were captured in natural lighting conditions with a blue background that was later replaced with other artificial background.

The exact date when sign language first appeared is unknown. However, with all of the spoken languages in existence, it is thought that hand gestures have existed for the same purpose as spoken language. Hand gestures and sign language arose from a need for a new way to communicate or engage.Although it is unknown when and where a deaf person first used sign motions, the first written sign language was discovered in Ancient Greece. The father of the deaf is called as ”l’Eppe.”

Among many approach made for the gesture recognition, sensor-based approach was the first. Sensors where used to capture the data from the gestures made. LED sensors, Microsoft Kinect are some of the sensors used. Along with these, vision based algorithms were also developed that used camera to capture the gestures made by the individual. The approach focused on capturing the primary features from the photographs and later it was used for the videos and real time captures as well. During the approach used many lighting circumstances, background noises limited the performance of the vision-based approach. Many strategies were studied that may reduce these effects to correctly recognize the signs. Algorithms such as CNN, SVM, HMM Open Pose Estimation and others were experimented with for the sign gestures recognitions.

Masood Et al. [3], Pandey et al. [4] all have used the basic approach of CNN model to recognize the hand gestures for the sign language. Another paper represents the work by the authors in creating a large scale word level dataset in American Sign Language(WLASL) [1]. The dataset contains the video of the word gestures where 2000 plus words gestures are performed by the participants. The authors aimed to make the dataset public to the research community so as to help expand the research work on the sign language. With the development of the dataset, authors were able to experiment with different deep learning methods for the word-level sign recognition and the performance all the used methods were evaluated. A holistic visual appearance based approach and 2D human pose based approach are implemented and compared.

Many other approaches were tested towards the recognition of the sign language hand gestures. Papers [5], Deep Learning for Alphabet Sign Language [6], Attention is all you need [7] are some of the research done in this field where the later one discusses on a totally new approach of using transformers that uses the concept of attention in the method to learn the sequences faster and effectively.

3. Methodology

3.1 Deep Learning Approaches

Convolutional Neural Networks (CNNs) were first developed over 20 years ago as the main machine learning approach for vision-based recognition, which is widely utilized in sign language recognition. Deep learning has become more common as hardware and software have improved. As new techniques are presented, the number of layers in the deep learning methodology is exceeded. However, as deep learning algorithms advanced, various input/ gradient vanishing problems emerged when the input was processed through numerous layers. These issues are being solved by new and contemporary architectures throughout time.

3.1.1 DenseNet

DenseNet gets its name from the fact that each layer in a DenseNet architecture is linked to the next. L layers have L(L+1)/2 direct connections. Each layer uses the feature maps from all previous levels as inputs, and its own feature maps as inputs for subsequent layers.As simple as it may appear, DenseNet effectively connects every layer to every other layer. This is the key idea, and it is really powerful. The input of a DenseNet layer is the concatenation of feature maps from previous layers. It is critical that the feature maps be concatenated before

they are combined. It's critical that the feature maps are of the same size. But we can't just keep the feature maps the same size across the network; convolutional networks use down-sampling layers to change the size of feature maps. To down-sample layers and ease feature concatenation in this design, the network is partitioned into many densely connected blocks, with the size of the feature maps remaining constant inside these dense blocks.

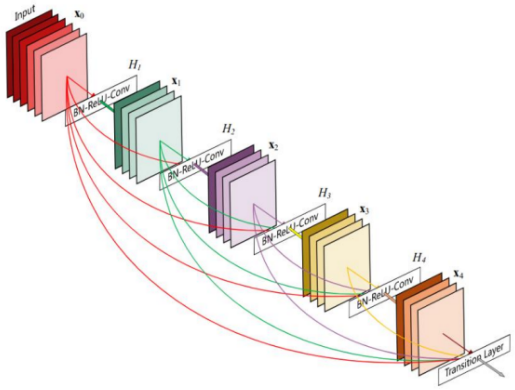

Figure 1: DenseNet Architecture

3.1.2 Transformer

Recurrent neural network was one of the best ways to process the sequence dependent data. A new model architecture began surpassing the RNN architecture since 2017. The paper "Attention is All You Need" [7] introduced this new architecture: Transformer. As the title of the paper suggest, the mechanism of using "neural attention" that does not use RNN or CNN could improve the translation tasks of sequence models. Transformer is a deep learning model the uses the mechanism of attention, differential scoring the importance of the features of the input data. Transformers have been proving very effective in sequence data and are the current state-of-art models for processing sequence data, most prominent in text processing and machine translation.

The transformer Figure 2 is divided into three components:

a. Encoder: Encodes an input sequence to state representation vectors

b. Decoder: Decodes the state representation vector to generate output

c. Attention Mechanism: Enables transformer to focus on the important features of the sequential input data which is used repeatedly within both encoder and decoder to contextualize input data.

Not all information fed to a model are equally important. Models should pay less attention to some features and more attention to some other features based on the importance of the features. The self-attention is calculated by creating three vectors,

Q is Query Vector, which contains the vector representation of one input in sequence K is Key Vector, which is vector representation of all inputs in sequence V is Value Vector, which represents of all values in the sequence

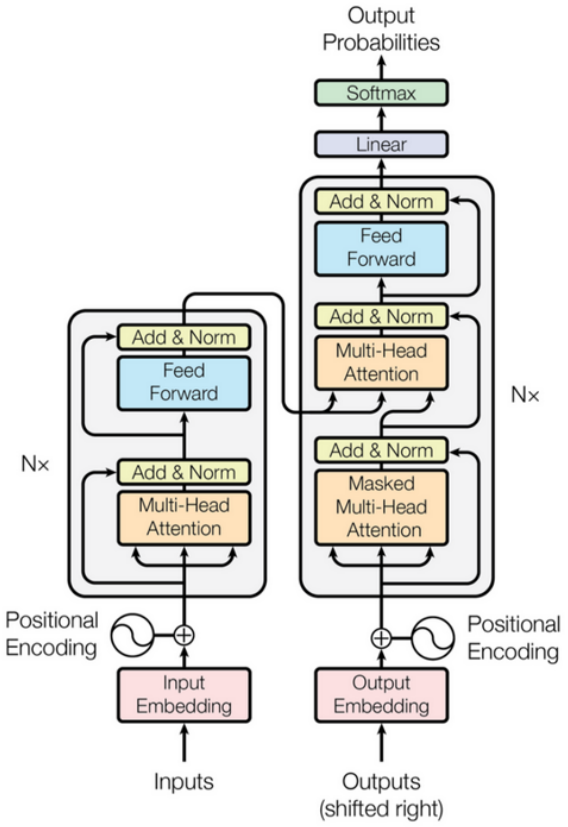

Figure 2: Architecture of Transformer

3.1.3 Long-Short Term Memory

The German researchers Sepp Hochreiter and Juergen Schmidhuber presented a variant of the recurrent net with "Long ShortTerm Memory Units" LSTMs as a solution to the vanishing gradient problem. LSTMs aid in the preservation of error that can be propagated backwards in time and layers. They allow recurrent nets to learn over multiple time steps (over 1000) by keeping a more consistent error, so establishing a route to link causes and effects remotely. LSTMs are specifically developed to prevent the problem of long-term dependency. They don't have to work hard

to remember knowledge for lengthy periods of time; it's like second nature to them. LSTMs has a key idea that is a cell state. It acts as a conveyor belt that is run throughout the entire chain of the network with very little linear interactions. Information can flow easily without being changed along the way.

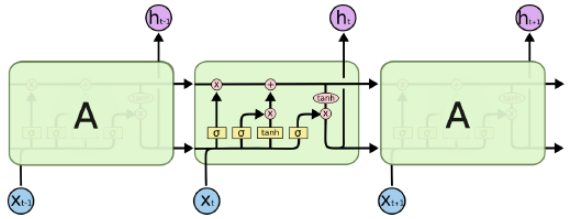

Figure 3: Module in LSTM with interacting layers

LSTM is unable to remove or add information to the cell state which is regulated by the structures called gates. Gates gives a way for the information to get through optionally and are composed of the sigmoid neural net layer with a point-wise multiplication operation. The output from this sigmoid layer is between zero and one that determines how much of the component is to be let through. Here, zero means nothing is let through whereas one means everything is let through.

3.2 System Model

The system model proposed is shown in Figure 4. As shown in the system block diagram, the initial stage in this work would be data gathering and dataset preparation. The dataset will be manually produced by taking- hand gesture video with the camera that is available with various subjects. The video sequence will be pre-processed. The video frames will then be enhanced to produce the frames variety in the dataset. The dataset will then be pre-processed using various approaches. The dataset will then be split into two parts: training and testing. After that, using the train dataset, a trained model is created, and the model is given the test dataset to predict the photos. The identified sets of photos will then be assessed and analyzed using several evaluation criteria, with the ultimate output being a written representation of the discovered indication.

During the data acquisition process, a mobile camera is used that captures the video sequences of the hand gestures made by the signer participants. Then, from the captured video sequences the frames are extracted and the extracted frames are processed further with different augmentation techniques to create even larger set of dataset.

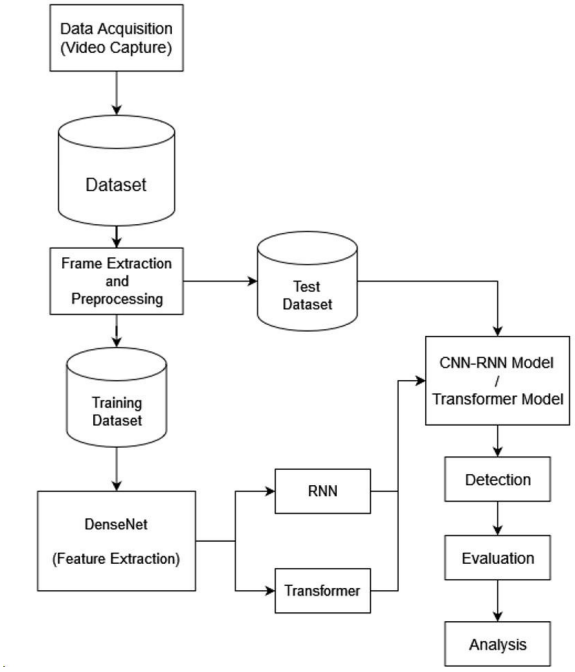

Figure 4: System Block Diagram

3.3 Dataset Description

With machine learning based classification the main problem is that it solely relies on the availability of datasets. For the collection of the data on word level hand gestures, I opted for the regular signer participants as well as the collection of videos from the internet. Initially, National Federation of the Deaf was contacted where they pointed out the copyright of the sign languages that is going to be used. To not violate the copyright of the sign languages, approval from the National Federation of the Deaf was taken. Then to actually collect the data, a school for the Deaf in Kathmandu was contacted. A letter requesting for the cooperation and coordination for the video data collection was prepared from the college.

As for the collection of data, video sequences for each word sign languages was taken manually from different i.e., students of +2 studying in Central Secondary School for the Deaf. 21 students participated during the video collection of the hand gestures for the words. 71 Words were taken during the video collection. Each word hand gesture was performed only once where some words required single hand or both hands. The video was captured during early morning hours in a confined small classroom of the school with a black backdrop.

Students were asked to wear a black t-shirt and gloves during the video capture. The distance between the camera and the participants was maintained 3 feet. During the video capture, students were shown the words in a play card printed in big fonts and students performed the gestures representing the word. The camera used was camera available in Iphone 12 pro max with resolution of 1920 by 1080p taken in 60 fps. The video captured were of 4-5 mins long with size ranging from 500-950 MB. The words that was taken during the video capture are as follows:

१. भोलि	२५. किताब	४९. खान्छु
२. आज	२६. घर	५०. पृथिवी
३. तपाइलाई	२७. गाडी	५१. गोलो
४. पर्सि	२८. स्याउ	५२. सुर्य
५. हामी	२९. आँप	५३. चन्द्रमा
६. तिम्रो	३०. अंगुर	५४. रुख
७. हाम्रो	३१. सुन्तला	५५. हिलो
८. तिमि	३२. केरा	५६. ऊ
९. मेरो	३३. मैले	५७. अल्छि
१०. नाम	३४. मलाई	५८. मेहेनेती
११. भेटौला	३५. मन	५९. राम्रो
१२. छ	३६. लाग्छ	६०. काम
१३. हो	३७. जान्छु	६१. गर्छ
१४. हुनेछ	३८. बिद्यालय	६२. धेरै
१५. बन्द	३९. बिदा	६३. प्रथम
१६. आइतबार	४०. हिमाल	६४. दोस्रो
१७. सोमबार	४१. देखियो	६५. भयो
१८. मंगलबार	४२. पानी	६६. मादल
१९. बुधबार	४३. पर्छ	६७. संग
२०. बिहिबार	४४. जाडो	६८. गुलाब
२१. शुक्रबार	४५. गर्मी	६९. बनायो
२२. शनिबार	४६. लाग्छ	७०. धन
२३. कापी	४७. हुन्छ	७१. रोपाई
२४. कलम	४८. खाना	

3.4 Preparing Video Frames

To prepare the video of the Nepali Sign Language, the videos were taken continuously and needed to be trimmed for each sign. Each sign language was trimmed from the continuous video to extract a separate individual sign for each of them. Each sign was about 1-3 seconds in length at 60 fps. For our model we required N (30) number of frames from each sign video. The frames were extracted by a simple algorithm so as to gather most of the key frames from the video. The pseudo-code for the algorithm is mentioned below:

Algorithm: *Frame Extraction*

input: *video, N no of frames*
output: *N frames*

```
video_length = video.frames.length()
max_seq = N
fskip = int((length-1) / max_seq)
f_index = int(video_length / 2) – int(max_seq * fskip / 2)
for j = 0 to 19 do
    current_frame = video.frames[f_index]
    new_frames[j] = current_frame
    f_index += fskip
end
return new_frames
```

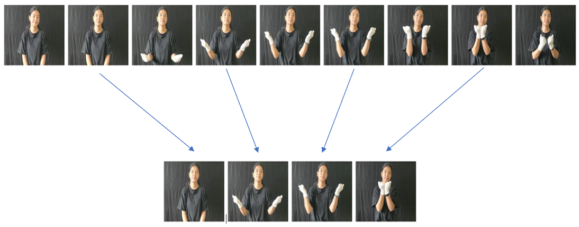

Figure 5: Frame Extraction

3.5 Analysis Metrics

To calculate the performance of the proposed system, we will use Accuracy, Precision, Recall, F1-Score. Accuracy is the closeness of measurement to a specific value, given as,

$$Accuracy = \frac{TP+TN}{TP+FN+FP+TN}$$

Precision is the fraction of correctly classified positive examples divided by the number of examples labeled by the system, given as,

$$Precision = \frac{TP}{TP+FP}$$

Recall, also known as true positive rate or sensitivity, is the probability that the model correctly identifies the anomaly detected.

$$Recall = \frac{TP}{TP+FN}$$

F1-Score is the harmonic mean of precision and recall. It gives the single measure of comparison and higher is better.

$$F1-Score = \frac{2*Precision*Recall}{Precision+Recall}$$

4. Results

The Nepali word level sign language recognition system was implemented and evaluated with two different approaches, (a) Using CNN-RNN model, (b) Using Transformer model. The formed dataset was split in 70:30 ratio. Here,

- Total hand gestures:9
- Total dataset in single hand gesture:105
- Total training dataset:765
- Total test dataset:180

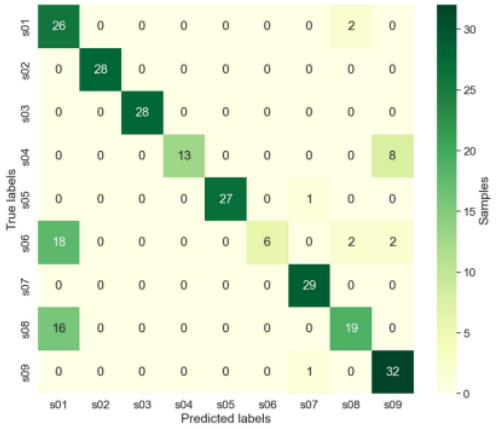

Figure 7: Confusion Matrix of CNN-RNN Model

CNN-RNN Model

For CNN-RNN model, DenseNet was used for feature extraction as feature extractor and the extracted features was fed to LSTM for sequential learning. The input to the model is sign language videos which are a collection of image frames. The video was processed to extract 30 frames from each video as input. The images from frames of the video were resized to 224 x 224. The sequential model with LSTM was trained using batch size of 32 with default learning rate of 0.001 of Adam optimizer. The model was trained for 70 epochs.

The results obtained are as mentioned:

The Figure 6 shows the performance metrices of CNN-RNN Model.

Accuracy: 0.8062015503875969

	precision	recall	f1-score
0	0.43	0.93	0.59
1	1.00	1.00	1.00
2	1.00	1.00	1.00
3	1.00	0.62	0.76
4	1.00	0.96	0.98
5	1.00	0.21	0.35
6	0.94	1.00	0.97
7	0.83	0.54	0.66
8	0.76	0.97	0.85

Figure 6: Performance Metrics of CNN-RNN Model

The overall accuracy of the model was 80.62%. It can be seen from the confusion matrix that in overall the model was able to correctly classify most of the signs, some of the signs were mis-classified like S06 was mis-classified as S01.

```
Test video path: aug_0_12p15.mp4
Frames: 37, Step: 1, Frames: 3-33
1/1 [==============================] - 0s 48ms/step
Predicted:  s12
--------------------------------
  s12: 94.93%
  s14:  3.17%
  s3:  0.74%
  s16:  0.47%
  s1:  0.43%
  s15:  0.21%
  s10:  0.02%
  s2:  0.02%
  s9:  0.01%
Frames: 37, Step: 1, Frames: 3-33
```

Figure 8: Correct Prediction

```
Test video path: aug_0_15p09.mp4
Frames: 83, Step: 2, Frames: 11-71
1/1 [==============================] - 0s 59ms/step
Predicted:  s1
--------------------------------
  s1: 38.99%
  s15: 38.19%
  s12: 11.60%
  s16:  6.46%
  s9:  1.59%
  s14:  1.46%
  s3:  0.77%
  s10:  0.74%
  s2:  0.20%
Frames: 83, Step: 2, Frames: 11-71
```

Figure 9: Incorrect Prediction

The Figures 8 and 9 show the incorrect and correct prediction respectively.

Transformer Model

For Transformer model as well, DenseNet was used for feature extraction as feature extractor and the extracted features was fed Transformer. The input to the model is sign language videos which are a collection of image frames. The video was processed to extract 30 frames from each video as input. The images from frames of the video were resized to 224 x 224 as well. The model was trained for 30 epochs.

The results obtained are mentioned below:

The Figure 10 shows the performance metrics of Transformer model.

```
Accuracy:  0.8914728682170543

              precision    recall  f1-score

           0       0.81      0.93      0.87
           1       0.90      1.00      0.95
           2       0.80      0.57      0.67
           3       1.00      1.00      1.00
           4       0.76      1.00      0.86
           5       1.00      0.74      0.85
           6       1.00      0.86      0.92
           7       0.85      1.00      0.92
           8       0.97      1.00      0.98
```

Figure 10: Performance metrics for Transformer Model

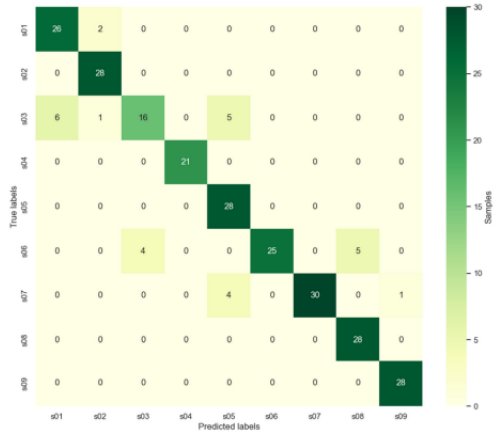

Figure 11: Confusion Matrix for Transformer Model

The overall accuracy of the model was 89.14%. Almost all of the signs were able to be correctly classified by the Transformer model.

The Figures 12 and 13 show the incorrect and correct prediction respectively.

```
Test video path: aug_0_12p15.mp4
Frames: 37, Step: 1, Frames: 3-33
1/1 [==============================] - 0s 60ms/step
Predicted:  s12
--------------------------------
  s12: 45.40%
  s9: 27.77%
  s14:  8.41%
  s1:  7.82%
  s3:  5.37%
  s10:  2.18%
  s16:  1.79%
  s15:  0.84%
  s2:  0.42%
Frames: 37, Step: 1, Frames: 3-33
```

Figure 12: Correct Prediction

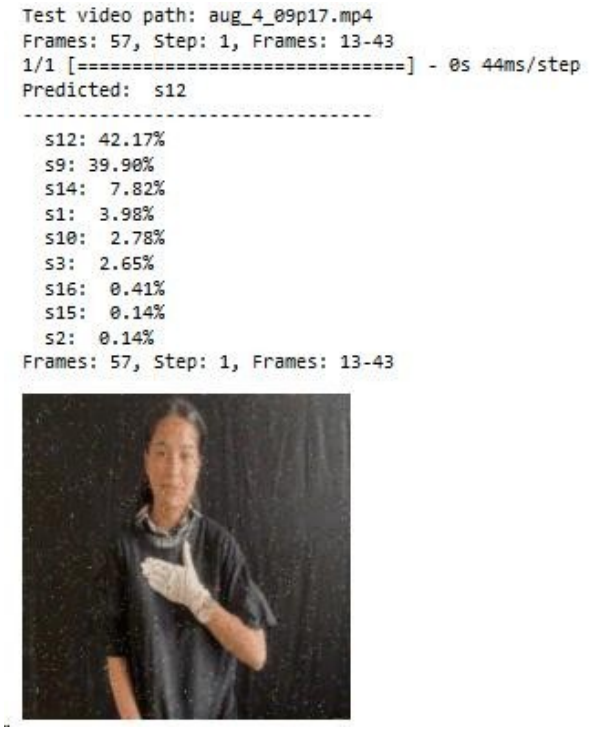

Figure 13: Incorrect Prediction

In both CNN-RNN model and Transformer model, we can see a little bit of overfitting of the model. Different methods such as regularization, dropout, learning rate variation, etc. were used, that helped improve the performance slightly. Both the approaches used are deep learning algorithms requiring large amount of data. As the model is too complex for the little data, they could not perform well and resulted to overfitting. To improve the performance, collection of more data was done from the online available data and different video augmentation techniques were used. The data still shortfalls for proper working of the system.

5. Conclusion

With the main idea of researching in the field of NSL where there has been very few research and experiments on this side of the language this work aimed for the use of the CNN architecture DenseNet in combination with RNN technique and Transformer to recognize the Nepali Sign Language in word level. This work was focused on gathering the videos of the hand gestures to prepare the dataset in word level for the Nepali Sign Language. The videos of 71 Nepali sign language gestures were captured and to increase the dataset video augmentation techniques were used. After the end of the dataset preparation total number of videos for the dataset was more than 9 thousand.

The evaluation of the word level Nepali sign language using both CNN-RNN model and Transformer model resulted in transformer model providing better performance. With a smaller number of sign-language gestures classes, the performance was better that with more classes. The best performance was obtained with 9 sign-language classes using 20 frames using Transformer model and was obtained to be 89.14%. For the entire dataset with 71 sign-language gestures, the best performance was given by transformer model using 30 number of frames and obtained to be 55.92%.

6. Acknowledgements

Authors express their deepest gratitude to National Federation of the Deaf Nepal, Bhrikutimandap for allowing the use of sign language for the research work. A special thanks to the Central Secondary School for the Deaf, Naxal for the coordination and allowing the participation of the +2 students to prepare the video dataset of the Nepali Sign Language. Sincere thanks to the +2 students studying in the school for their enthusiastic participation and cooperation during the data collection and to everyone providing possible help with the technical understanding of the approach proposed and to resolve the technical difficulties during the implementation of the work.

References

[1] Asl alphabet — kaggle.

[2] A comprehensive guide to convolutional neural networks — the eli5 way — by sumit saha — towards data science.

[3] Sushila Sipai. Nepali sign language translation using convolutional neural network. I:49–53, 2018.

[4] Brazilian sign language - words recognition — kaggle.

[5] Marina Sokolova and Guy Lapalme. A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45:427–437, 7 2009.

[6] Rangel Daroya, Daryl Peralta, and Prospero Naval. Alphabet sign language image classification using deep learning. *IEEE Region 10 Annual International Conference, Proceedings/TENCON*, 2018-October:646–650, 2 2019.

[7] Ashish Vaswani. Attention is all you need arxiv:1706.03762v5. *Advances in Neural Information Processing Systems*, 2017-Decem:5999–6009, 2017.