International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 06 Issue: 05 | May 2019 www.irjet.net p-ISSN: 2395-0072

American Sign language Classification

Wenhui Liang[1], Dixita Bhanderi[2]

[1]Graduate student, Department of Computer Science, California State University, Sacramento
[2]Graduate student, Department of Computer Science, California State University, Sacramento

---

Abstract - *There is an undeniable communication issue with the hearing community and deaf minority. Innovations in automatic sign language recognition attempt to tear down this barrier obstruction. This paper discusses an American Sign Language (ASL) alphabet gestures' image recognition system. Images of the gestures are classified into their corresponding individual English alphabets, which are further converted into that alphabet voice. We achieved this work by using the convolutional neural networks (CNNs) and transfer learning with VGG16 for gestures classification and used gTTS google library for converting alphabet to audio. Our implemented CNN model gives considerable performance with 0.99 precision. Which can be further trained with larger dataset and solution is CNN model and transfer learning approach for ASL classification and discusses the performances' achieved with experimental results.*

Keywords **-** ASL, ASL Classification CNN, Transfer Learning, Sign Language Classification, Gesture recognition, gTTS.

1.INTRODUCTION

Gesture and sign language recognition is an integral part for an efficient communication between the hearing and hard-of-hearing(deaf) population. The method that has been adopted since years was to facilitate communication with the help of a human sign language expert. However, this is a primitive solution which is highly inefficient and expensive as not everyone around us are trained in sign language interpretation. Therefore, an efficient automatic translator was to be designed which would facilitate the deaf community to communicate with others either through text or sound depending on the end user.[8] Researches have been advancing in this area since the past decade and with the advent of newer emerging technologies there has been rapid progress in results with respect to accuracy.

American Sign Language (ASL) substantially facilitates communication in the deaf community. However, there are only ~250,000-500,000 speakers which significantly limits the number of people that they can easily communicate with [7]. Hence, we choose to work on the American Sign Language gesture image classifier to implement few basic and robust classification approaches and functionalities needed to put on together.

For the classification of ASL images, we implemented an image pre-processing module, which pre-processes dataset images and splits the dataset into training and testing portions. Secondly, we built a classification module for the labelling of the signs.

For this purpose, we used CNN algorithms as its highly recommended models for the image classification. CNN models do feature extraction by itself. However, with this approach, we need large dataset with variety of cases while training the model in order to receive good accuracy. Hence, we used the dataset which fulfils those needs for training purpose.

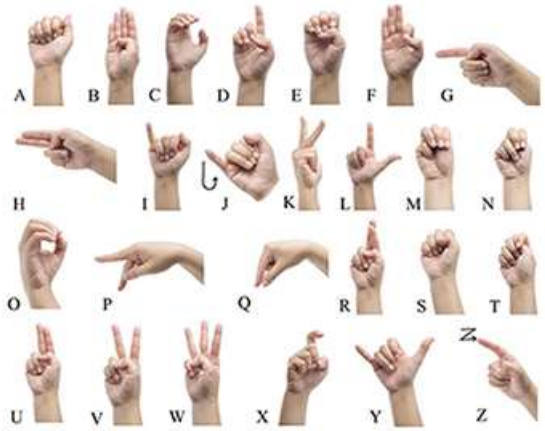

Fig -1: American Sign Language Alphabets

For more better performance of our prediction module we implemented transfer learning approach to gain previously trained models with larger datasets. We used VGG16 model for this purpose. Finally, our classification module classifies the unknown test images for the verification of the results. Finally, text-to-speech module was implemented to play the sound of classified letters from the generated mp3 files.

From the classification perspective, this problem represents a significant challenge due to many considerations, including:

- Gesture image background (e.g. different backgrounds, background depths)
- Image quality (e.g. blurred sign)
- Hand gesture captured angle

This paper first outlines problem formulation where we mentioned the overview of the approach and task performed in order to solve a problem and what way we implemented it. Next, we focused on system architecture

and detailed description of the various algorithms we implemented and show comparison with charts and displayed the confusion matrix for best achieved model. After that we discussed the experiments, we performed with the real-world dataset, the dataset other than the one we used for training and results of the best model and transfer learning. At last we have mentioned our related work and task division and we concluded our work.

2.PROBLEM FORMULATION

Our problem consist of three task to be perform:

1. Obtaining sign image from the user
2. Classify that sign image to a letter
3. Playing the sound of the particular alphabet

Our classifier features a pipeline of image to audio. It that takes image of an ASL gesture through notebook, than we classify that using classification module into letter using our best CNN model achieved. Finally, we use our text to speech module in order to play the alphabet.

3.ALGORITHM DESIGN

3.1 System Architecture

Our system is consisting of three different modules:

1. Image pre-processing module
2. Classification module
3. Text-to-speech module

Those modules take input from its previous module and returns output which is used by next module. Our classification module plays an intermediate and important role. It is a core of the system which does all the major functions.

3.2 Image pre-processing module

This module is implemented for the pre-processing of the dataset images. It performs multiple tasks, such as loading the image data from the dataset, resizing images, image to tensor creation, and splitting that into train and test datasets. This is the stage where image data gets ready to fed into the classifier models for training. Algorithm Description: Initially it loads the list of all images from the dataset with all label information and converts those images into NumPy array, and then using pandas library function it performs one-hot encoding on the labels. Later, that tensor split into train and test sets of 70-30 ratio using Sklearn's train_test_split library function.

3.3 Classification Module

This is the module which does the actual job and our most experiments have been done in this module. It consists varieties of classification models and evolvement of them into our best model for our purpose of classification of ASL. As we choose to perform the classification using CNN models, we implemented various models of CNN with different number of layers, different optimizer and activation and achieved the final best CNN model for the classification. All the CNN models we implemented with Early Stopping, ModelCheckPoint and accuracy as performance matrix in order to get better weights saved while training the models.

1) CNN Model 1 Algorithm Description: As an attempt to determine the workflow of our previous module for image pre-processing, we here implemented a basic approach for the CNN model building and training for image classification. We used two Conv2D layers with 32 neurons followed by Maxpooling2D layers, a dropout layer with adam optimizer and ReLU activation. This model gave the accuracy of 0.52%.

2) CNN model 2 Algorithm Description: In order to improve our CNN model 1 for better accuracy, we added one more Conv2D layers with 64 neurons followed by Maxpooling2D layer, a dropout layer with adam optimizer, ReLU activation and steps per epoch as 10. This model gave the accuracy of 0.63%.

3) CNN model 3 Algorithm Description: Extending the CNN model 2 approach, in this model we performed parameter tuning on optimization and activation values. We received improved accuracy of 69%.

4) CNN model 4 Algorithm Description: In this model we tuned other parameters of the model. We changed the value of the parameter 'steps_per_epoch' in the fit_generator function. We followed below formula for the value of steps per:

steps_per_epoch = total_records(train+test)/ batch size

With this model we received improvement in the accuracy significantly. Accuracy as 98% and f1-score of 0.98. We considered this model as our best model.

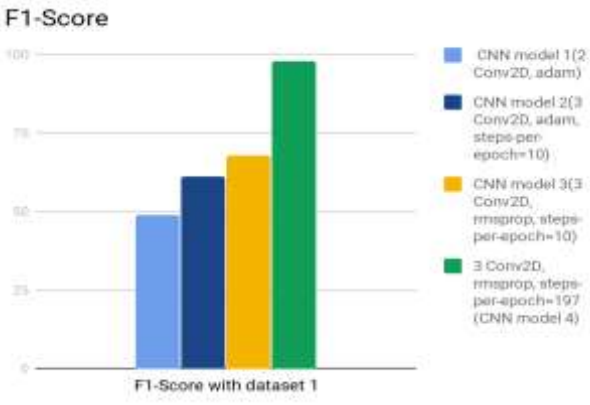

Chart -1: Performance evaluation with various parameters for CNN model with dataset 1

We compared all these four models directed our implementation of the next model towards the parameters which fits best with model. We used precision score and accuracy matrix to compare the performance of each model. We generated the graph of precision score vs each of the four models which can be see Figure 2. We can here, as we added one more Conv2D layer and a dropout layer the performance increased, and as we changed the optimizer from adam to rmaprom performance again increased. We also used sgd optimizer for that model but we got better result with rmsprop. Finally, when we changed our steps_per_epoch parameter with the proper value from formula, the model achieved significant precision score. As we can see in the confusion matrix of that model in Figure: 3, there are few miss classifications for letter p as letter s other than that letters are classified correctly.

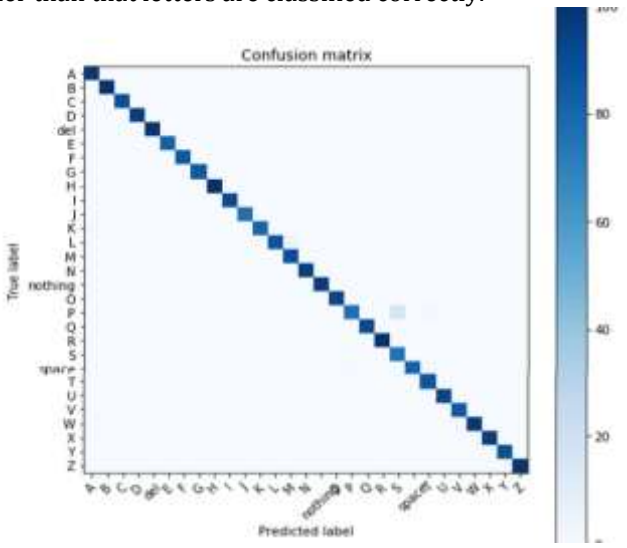

Fig -2: Confusion matrix for CNN model 4 (best model)

3.4 Text-to-Speech Module

This module performs the text to voice task. It takes input of the alphabets from the classification module and plays the sound of that alphabets as voice.

1) Algorithm Description: We implemented this by using google library gTTS which converts text into speech. It takes classified image label, alphabet as text and saves the audio file which later played using the play function of the MediaPlayer fuction of the vlc library.

4.EXPERIMENTAL EVALUATION

4.1 Methodology

We used the ASL dataset provided by Kaggle. It is has a very large dataset consists of 3000 images for each alphabet signs, with difference in background and various camera angles. Moreover, this dataset has the images for three different scenarios other than the actual hand gesture of the sign. To be specific, the images with nothing in it no hand gesture just any random backgrounds, images with only hand no actual signs or wrong signs or gestures and images for space. Space as in space between letters.

In terms of the size, this dataset is very large to handle, (3K*29) images in total, we decide to take only 1000 images for each letter (1K*29) to use for our model training and experimental purpose in order to perform this classification.

For the further experiments and testing, we used another Kaggle ASL data set called real world images which contains 30 images for each alphabet. We used that dataset to verify our best CNN model achieved and we received less accuracy of 19% and f1- score 0.18.

In order to improve the performance with our second dataset we decide to use the transfer learning. For the transfer learning model implementation, we used VGG16 model.

After receiving the lower accuracy of our transfer learning model compare to our best model using this second dataset, we decided to train our best model and our transfer learning model with the second dataset. In that experiment, we received 64% and 49% accuracy and f1-score 0.64 and 0.33 respectively for those models Figure. 4.

4.2 Results

We received similar performance for our first dataset of 29K images using both the models, our best model and transfer learning model. However, we did not receive better performance using second dataset, holding images with noisy and deep backgrounds using our both the models. Our best model performed well in compared to our transfer learning model with the f1-score of 0.64 and 0.33 respectively Figure. 5.

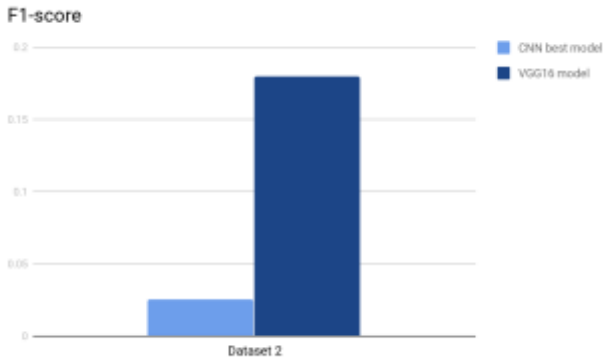

Chart -2: Comparison of CNN best model and VGG16 Transfer Learning Before training with dataset 2

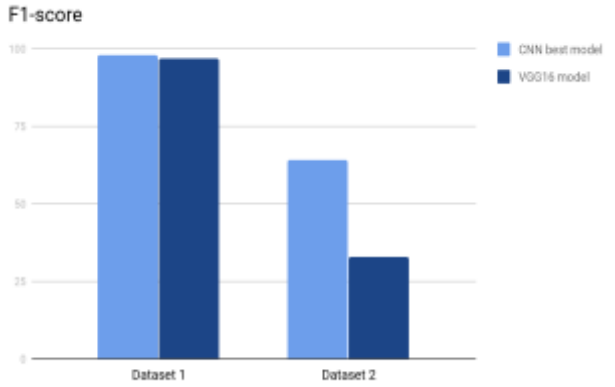

Chart -3: Comparison of CNN best model and VGG16 Transfer Learning after trained with dataset 2

We can compare the differences in graphs for Accuracy and loss vs epoch while model training for Dataset1 & Dataset 2.

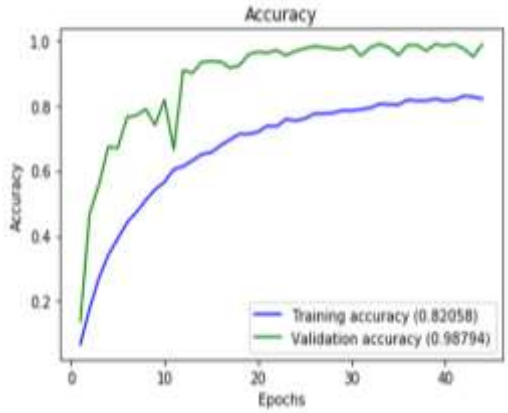

Chart -4: Epoch vs. Accuracy for CNN best model with dataset 1

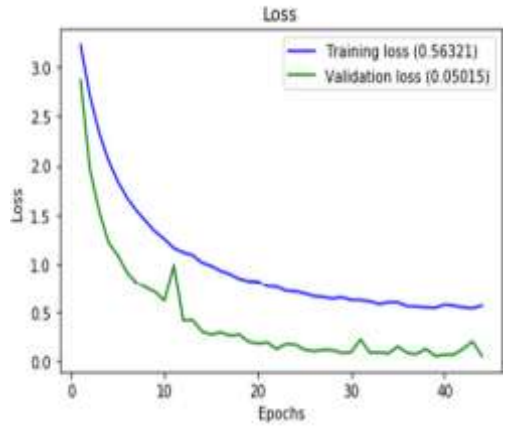

Chart -5: Epoch vs. Loss for CNN best model with dataset 1

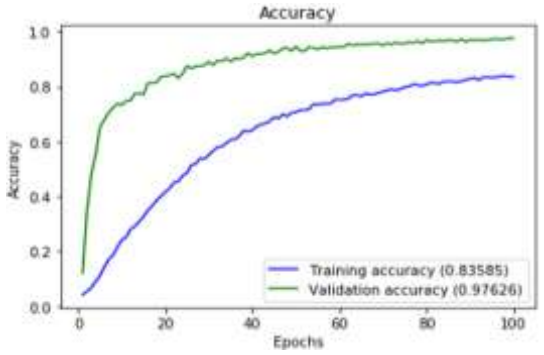

Chart -6: Epoch vs. Accuracy for VGG16 transfer learning model with dataset 1

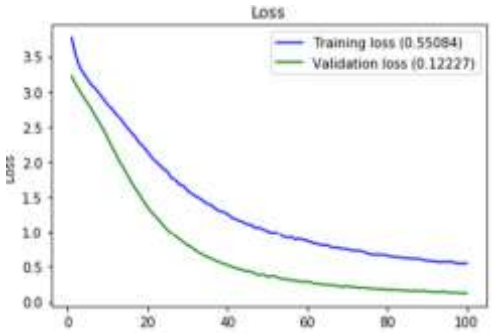

Chart -7: Epoch vs. lose for VGG16 transfer learning model with dataset 1

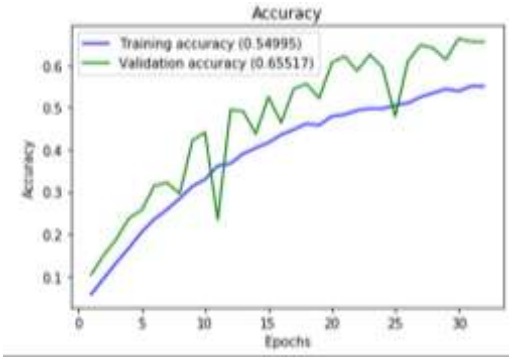

Chart -8: Epoch vs. Accuracy for CNN best model with dataset 2

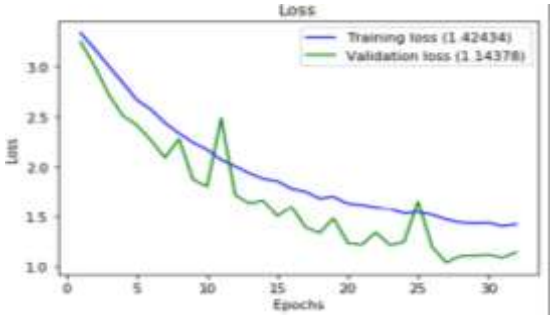

Chart -9: Epoch vs. loss for CNN best model with dataset 2

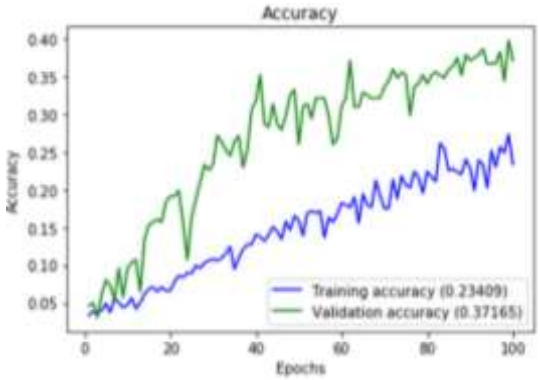

Chart -10: Epoch vs. Accuracy for VGG16 transfer learning model with dataset 2

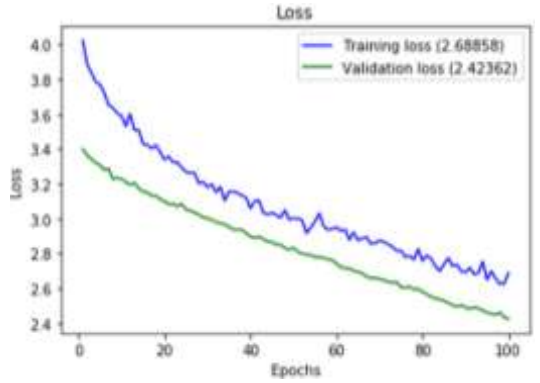

Chart -11: Epoch vs. loss for VGG16 transfer learning model with dataset 2

5.RELATED WORK

The work of ASL sign language translation has been done for the past couple of decades. Due to the recent advancement in the machine learning technology, the accuracy of the image classification of ASL translation has approached 99.99%. Various techniques are used to improve the accuracy of ASL translation, some leverage the time series information in the fingerspelling of words [6].

Convolutional Neural Networks have been amazingly effective in image recognition and classification issues, and have been effectively implemented for human gesture recognition lately. Specifically, there has been work done in the domain of communication via gestures recognition utilizing deep CNNs, with info recognition that is sensitive to something beyond pixels of the pictures. With the utilization of cameras that sense depth and shape, the procedure is made a lot simpler through developing characteristic depth and movement profiles for each sign gestures movement [3].

As of not long ago, however, strategies for automatic gesture-based language recognition couldn't utilize the depth detecting technology that is as widely available today. Past works utilized very basic camera technology to create datasets of simply pictures, with no depth or shape data accessible, simply the pixels present. Attempts at utilizing CNNs to deal with the task of classifying pictures of ASL letter signals have had some success [5], yet using a pre-prepared GoogLeNet architecture.

Paulo Trigueiros, Fernando Ribeiro, and Luís Paulo Reis [12] have proposed a constant vision-based framework whose intention is to perceive Portuguese sign-based language. They utilized Kinect Camera to remove hand features. For model training and Gesture classification open source Dlib library was utilized, a broadly useful cross platform C++ library equipped for SVM multiclass arrangement.

Haitham Hasan, S. Abdul-Kareem [9] have proposed a system for hand gesture recognition dependent on shape analysis. They utilized neural network-based way to classify among six static hand gestures to be specific open, close cut, glue, maximize and minimize. They have utilized a remarkable multi-layer perception of neural network for classification using back-propagation algorithm. They had the capacity to accomplish an accuracy of 86.38%.

Neha V. Tavan, Prof. A.V. Deorankar [10] in their work executed an algorithm to extract HOG features. These features were then used to train a artificial neural network which was later used with the end goal of gesture recognition.

Lionel Pigou, Sander Dieleman, Pieter Jan Kindermans, Benjamin Schrauwen [11]. Their contribution considers a recognition framework utilizing the Microsoft Kinect, GPU acceleration and convolutional neural networks (CNNs). Rather than building complex handcrafted features, CNNs can automate the procedure of feature development. They had the capacity to recognize 20 Italian motions with high accuracy. Their prescient model had the capacity to generalize training with a cross-validation precision of 91.7%.

The utilization of depth-sensing technology is rapidly developing in popularity, and different tools have been consolidated into the procedure that have demonstrated successful. Improvements, for example, specially designed colour gloves have been utilized to encourage the acknowledgment procedure and make the element extraction step increasingly productive by making certain gestural units simpler to distinguish and order [4].

6.CONCLUSION

We implemented and trained an American Sign Language classifier on a notebook using CNN algorithms and transfer learning VGG16 model. We are able to produce a robust model for all alphabets a to z, space and nothing labels. Because of the lack of variation in our dataset 1, the validation accuracies we observed during training were not directly reproducible upon testing on the dataset 1. We

hypothesize that with additional data, which we removed from the dataset 1 holding remaining 2000 images, the models would be able to generalize with considerably higher efficiency and would produce a robust model for other real world datasets holding background noise or depth.

Reference

[1] Barczak, A.L.C., Reyes, N.H., Abastillas, M., Piccio, A., Susnjak, T., A new 2D static hand gesture colour image dataset for ASL gestures, Research Letters in the Information and Mathematical Sciences, 15, 12-20, 2011.

[2] X. Chen and A. Yuille. Articulated pose estimation by a graphical model with image dependent pairwise relations. In Advances in Neural Information Processing Systems (NIPS), 2014.

[3] Agarwal, Anant & Thakur, Manish. Sign Language Recognition using Microsoft Kinect. In IEEE International Conference on Contemporary Computing , 2013.

[4] Cao Dong, Ming C. Leu and Zhaozheng Yin. American Sign Language Alphabet Recognition Using Microsoft Kinect. In IEEE International Conference on Computer Vision and Pattern Recognition Workshops , 2015.

[5] Garcia, Brandon and Viesca, Sigberto. Real-time American Sign Language Recognition with Convolutional Neural Networks. In Convolutional Neural Networks for Visual Recognition at Stanford University, 2016.

[6] Lifeprint.com. American Sign Language(ASL) Manual Alphabet (fingerspelling) 2007.

[7] Mitchell, Ross E., et al. "How many people use ASL in the United States? Why estimates need updating." Sign Language Studies 6.3 (2006): 306-335.

[8] Byeong Keun, Subarna Tripathi, and Truong Q Nguyen. Real-time sign language fingerspelling recognition using convolutional neural networks from depth map. In 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR), pages 136–140. IEEE, 2015.

[9] Hasan, Haitham Sabah, and Sameem Binti Abdul Kareem. "Gesture feature extraction for static gesture recognition." Arabian Journal for Science and Engineering 38.12: 3349-3366, 2013.

[10] Tavari, Neha V., and A. V. Deorankar. "Indian Sign Language Recognition based on Histograms of Oriented Gradient." International Journal of Computer Science Information Technologies 5 2014.

[11] Escalera, S., Bar, X., Gonzlez, J., Bautista, M.A., Madadi, M., Reyes, M., Ponce, V., Escalante, H.J., Shotton, J., Guyon, I.: Chalearn looking at people challenge 2014: Dataset and results. In: ECCV Workshop 2014.

[12] Trigueiros, Paulo, Fernando Ribeiro, and Luís Paulo Reis. "Vision-based sign language recognition system." World Conference on Information Systems and Technologies Madeira, Portugal. 2014.