Sign Language Translator using Deep Learning

Ashutosh Upreti
2014B4A70784G

Himanshu Singhvi
2014B2A70716G

Mehul Garg
2014B4A70805G

***Abstract*— A real-time sign language translator is an important milestone in facilitating communication between the deaf community and the general public. We present an American Sign Language (ASL) finger-spelling translator based on a convolutional neural network approach. We utilize a pre-trained VGG architecture trained on the ImageNet dataset, as well on our own dataset in order to apply transfer learning to this task. We have made minor changes to the dataset so as to make our model more robust by introducing new classes for dynamic ASL gestures.**

I. INTRODUCTION

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. The alternative of written communication is cumbersome, impersonal and even impractical when an emergency occurs. In order to diminish this obstacle and to enable dynamic communication, we present an ASL recognition system that uses Convolutional Neural Networks (CNN) in real time to translate a video of a users ASL signs into text.

Our problem consists of 3 tasks to be done :

- **Localiser**: Locating the users hand from the video frame
- **Recogniser** : Classifying each frame to a letter
- **Language Model** : Reconstructing and displaying the most likely word from the classification scores

From a computer vision perspective, this problem represents a significant challenge due to a number of considerations, including:

- Environmental concerns (e.g. lighting sensitivity, background, and camera position)
- Occlusion (e.g. some or all fingers, or an entire hand can be out of the field of view)
- Sign boundary detection (when a sign ends and the next begins)
- Co-articulation (when a sign is affected by the preceding or succeeding sign)

Our system features a pipeline that takes video of a user signing a word as input through a web-cam. We then extract individual frames of the video and pass it to our Recogniser module where CNN is used to classify the gestures. Finally, as real-time scenario will include a lot of noise, mis-classification of letters will take place occasionally. Therefore, to rectify this, we use a language model in order to output the most likely word to the user.

II. OUR APPROACH

A. *Localiser*

Our first task is to locate the user's moving hand from the video frame. To perform this localisation, we have taken help of the OpenCV library. A green band is tied as a wearable to the wrist and using color detection algorithms in OpenCV, we have segmented out the hand from the frame by creating a bounding box around the hand. The cropped image (bounding box) is then passed to our Recogniser module for image classification.

B. *Recogniser*

1) *Convolutional Neural Network (CNN):*

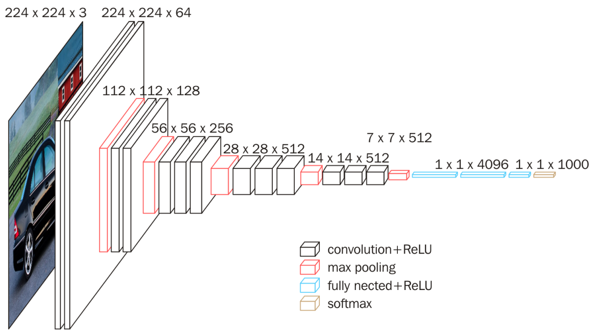


Fig. 1. VGG-16 architecture

Our ASL letter classification is done using a convolutional neural network. CNNs are machine learning algorithms that have seen incredible success in handling a variety of tasks related to processing videos and images.

A primary advantage of using CNN is it's ability to learn features as well as the weights corresponding to each feature. Like other machine learning algorithms, CNNs seek to optimize some objective function, specifically the loss function. We utilized a softmax-based loss function.

Using a softmax-based classification allows us to output values akin to probabilities for each ASL letter. These probabilities afforded to us by the softmax loss allow us to more intuitively interpret our results and prove useful when running our classifications through a language model.

2) *Transfer Learning:*

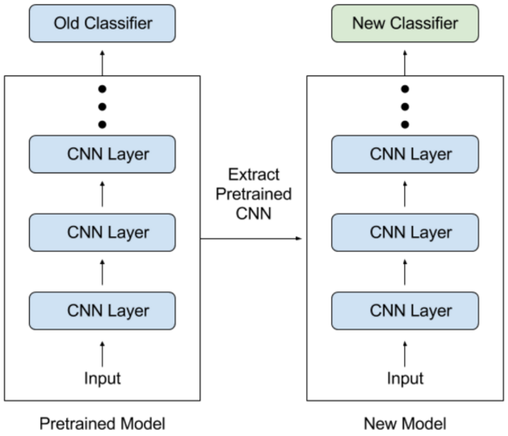


Fig. 2. Transfer Learning

Transfer Learning is a machine learning technique where models are trained on larger data sets and refactored to fit on more specific data. This is done by recycling a portion of weights from the pre-trained model and reinitializing or otherwise altering weights at shallower layers. The primary benefits of such a technique are its less computational time and less data requirements.

Our overarching approach here is to fine-tune an existing pre-trained model (VGG16) concatenated with few dense layers of our own. We have kept some of the earlier layers fixed (due to overfitting concerns) and only fine-tuned the higher layers of the network.

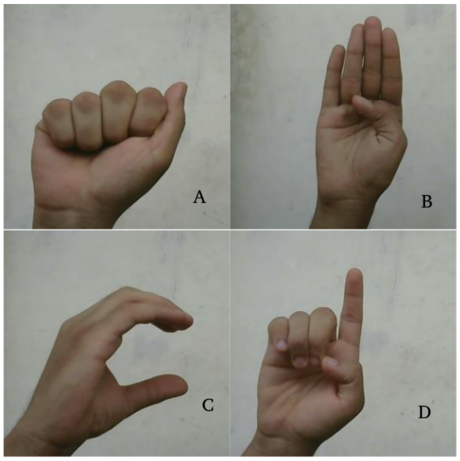
After training process is completed, multi-class classification of hand signals is performed. The predicted letters are concatenated to form a word. To indicate the completion of a word we have introduced a special gesture in our dataset. The final concatenated word is then passed to our language model to perform necessary error corrections and output the most likely word to the user.

C. *Language Model*

The final task of our problem is to output the desired word. The predicted word from our Recogniser module is processed by our language model. Due to the challenges mentioned in the Introduction section like Environmental concerns and Occlusion, there is a high probability that mis-classification of letters take place. Therefore, it is crucial for our language model to effectively perform functions like auto-correct and auto-complete.

III. Experimental Setup

A. *Dataset Information*



- Data Generation : Constructed an ASL dataset using OpenCV and a Webcam.
- Dataset consists of 100 images per class (letters)
- Apart from just letters, dataset consists of 2 additional classes:
  - None : Images with plain white background
  - Spell : Sign to indicate end of a word which is then autocorrected and spelled out.

B. *Results and Analysis*

We use a confusion matrix, which is a specific table layout that allows visualization of the performance of our classification model by class. This helps us evaluate which letters are mis-classified and draw insights for future improvement. The confusion matrix (Fig.4) shows that our model works well for almost all the classes.

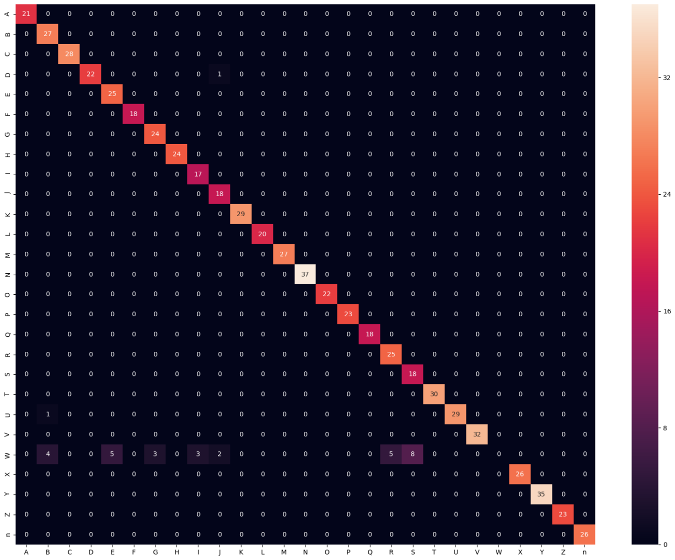


Fig. 3. Confusion Matrix

Graphs for Accuracy Vs Epochs (Fig.5) and Loss Vs Epochs (Fig.6) help us understand the rate of convergence of our model. They prove that due to transfer learning the model has converged extremely fast with above 99% accuracy.

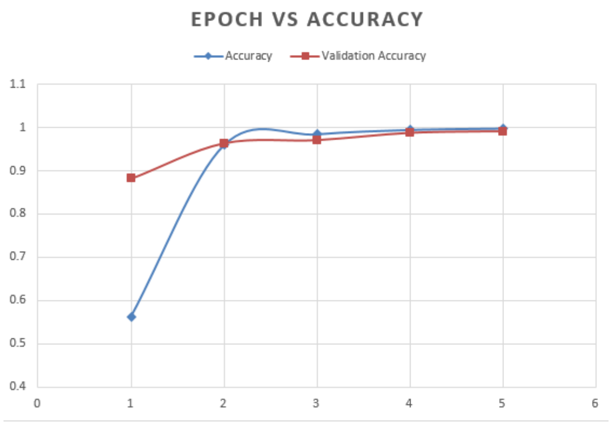


Fig. 4. Accuracy Vs Epochs

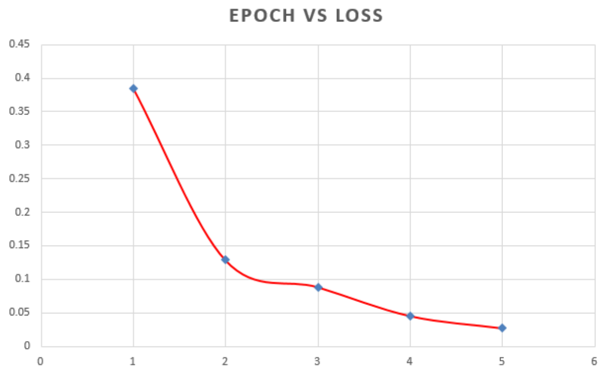


Fig. 5. Loss Vs Epochs

IV. CONCLUSION

- We have implemented and trained an American Sign Language translator on a CNN classifier using transfer learning
- Transfer Learning gave us a big boost to our model's accuracy (99%) and reduced the training time drastically (Refer Fig.5 and Fig.6)
- We hypothesize that with additional data taken in different environmental conditions, the model would be able to generalize better and would produce a robust model for all letters.

V. FUTURE WORK

- Additional features for the language model:
  - Predictive Typing
  - Grammar Checks
- Improvement to the localization module : Using better hand segmentation algorithms
- Constructing a smaller model for faster real-time prediction
- Using a bigger dataset with different backgrounds.
- Mobile Application : The full potential of what we are building can be best utilized by a portable device such as a mobile phone. Hence, building an easy to use interactive mobile app with our machine learning model running in the back-end would be a breakthrough in the vocally impaired community.

VI. ACKNOWLEDGMENT

We would like to extend our gratitude to Mr Sreejith V, Instructor In-charge of the course Software Development for Portable Devices, for his guidance and support.

References

[1] Houssem Lahiani, Mohamed Elleuch, Monji Kherallah, Real Time hand gesture recognition system for android devices.

[2] Brandon Garcia,Sigberto Alarcon Viesca, Real-time American Sign Language Recognition with Convolutional Neural Networks.

[3] Ryan White,Ali Farhadi, David Forsyth, Transfer Learning in Sign language.

[4] Shuying Liu,Weihong Deng, Very deep convolutional neural network based image classification using small training sample size.

[5] Lionel Pigou, Sander Dieleman, Pieter-Jan Kindermans, Sign language recognition using Convolution Neural Networks.