Eight or Three?: Multi-Language Sign Translation System

Amiy Yadav, Spencer Wilson

University of Colorado Boulder

Abstract. Imagine a world with universal real-time translators, even supporting signed languages. Here, we provide an exploration of sign translation between American Sign Language (ASL) and Indo-Pakistani Sign Language (ISL) over signed numbers. Final accuracy measures suggest perfect prediction in translation of ISL to ASL and 95% accuracy in translating ASL to ISL. Given an unlabeled image of a signed number in either ASL or ISL, determine the language, then determine the semantic, and finally translate the sign to the opposite language.

Keywords: Neural Networks, Sign Language Recognition, Sign Language Translation, VGG19

1 Introduction

There are around 70 million people in the world that are deaf, or hard of hearing, and fluent in a sign language [1]. There are nearly 300 sign languages in the world with the most used being Chinese (CSL or ZGS)(20 million user), Brazilian (BSL)(3 million users), and Indo-Pakistani (ISL)(1.8 million users) [2]. We also note that American Sign Language (ASL) is the third most used language in the USA after English and Spanish [3]. It is observe that the same signs have different meanings in different sign languages (alike spoken languages): e.g. the ISL sign for eight is the ASL sign for three, e.g. the ASL sign for two is a quite vulgar hand gesture in the United Kingdom. Our goal is to create a sign language translation system that does not require the input sign language to be labeled. We will begin with a bi-directional translation system between ASL numbers and ISL numbers. While we begin this research over numbers, future work shall explore sentence level translations. This can propagate into a greater research context to build virtual or robotic assistants capable of translation between multiple popular signing languages in real time, taking in a live stream of signing video, understanding the current language, and returning a signing video in the user-preferred sign language. This may encourage further exploration of the language by people with unaltered hearing and will allow for better communication between users of different signed languages.

There has been significant work in Sign Language Recognition (SLR) to written and spoken languages over the past 20 years, ranging from state vector machines (SVM) and Shift Invariant Feature Transformation (SIFT), and has seen recent precision advancements by using Convolutional Neural Networks (CNNs) [4]. Researchers have also experimented with data augmentation techniques which used wearable sensory modalities such as sensory gloves for recording different signs, transforming gestures into skeletal model which provide precise inputs, but these are time consuming and require a high computational capacity [4]. To our knowledge, there have not been efforts to integrate a language classifier to this translation pipeline to enable universal sign language translations. There have only been efforts to translate between a known sign language into a known written language, and from that known written language to a known sign language.

We propose an exploration of ISL and ASL signed numbers using various preprocessing techniques and feature extraction techniques to perform sign language classification, sign semantic classification, and sign language translation [5] [6]. Through experimentation with CNN methods we found optimal precision results of our models. The novel approach of language classification first will allow for future expansions to more languages for better user interfaces (UI) for deaf-mute users of the eventual virtual or robotic assistant.

2 Related Work

2.1 Image Acquisition and Pre-processing

Previous work experiments with different image pre-processing steps such as Scale Invariant Feature Transformation (SIFT) , Speeded-Up Robust Features (SURF), Histogram of Oriented Gradients (HOG), Canny Edge Detector, and Elliptical Fourier Descriptor [7] [8] [9] [10] [11]. The work suggests that HOG outperformed SIFT and SURF for sign language recognition [8]. Additional work notes that Canny Edge detection can integrate with HOG for improved results [10]. Finally, the literature suggests that elliptical Fourier Descriptor can be applied to improve feature extractions [11]. We propose an exploration of combinations of Canny Edge detector, multilevel HOG, and elliptical Fourier descriptor for optimal feature extractions from our scant datasets.

2.2 Sign Language Recognition

There have been many artificial intelligence (AI) methods applied to this domain in the past couple of decades, over time we have seen an evolution in automated sign language recognition and translation from traditional machine learning techniques like support vector machine (SVM) and k-nearest neighbour (kNN) toward deep learning methods like CNNs, recurrent neural networks (RNNs), skeletal model for static images, and Hidden Markov Model (HMM) for video translation [4] [12]. The five major signing components a machine must identify in order to understand a sign semantic are as follows: hand-shapes, location of hand, palm orientation, movement of hand, and facial expression [4]. Current work focuses on translation from one specified sign language to English, then from English out to another specified sign language [13] [7]. We propose an exploration of the VGG19 CNN pre-trained on ImageNet to construct a sign language translation architecture.

2.3 Language Classifier

Classification of textual languages has been done numerous times previously using various methods which included sentiment analysis, natural language inference, question answering and news categorisation [14] [15]. These classifications mainly revolved around CNNs and RNNs as their deep learning models [14]. Sometimes, researchers also included attention mechanisms or even a hybrid of aforementioned three methods [14]. Work has included interpreting multiple written languages in the same architecture [15]. But, there has been no significant research done on classification of multiple sign languages in the same architecture. We propose an exploration of multiple sign language classification prior to semantic classification to construct a multi-sign-language translation system.

3 Methods

3.1 Image Pre-processing and Feature Extraction

We performed the following pre-processing steps on ASL and ISL number images zero through nine using pre-existing collections for ASL and ISL in an attempt to gain higher accuracy from our models [6] [5]. An example of the original images is provided in Fig. 1 for your reference. For each original image we apply Canny Edge Detection to create new images. This pre-processed image set is then processed again using Histogram of Oriented Gradients (HOG). Finally, for each image pre-processed with Canny Edge Detection and HOG, we apply elliptical Fourier Descriptor to create the final pre-processed images (we term this CHF for convenience). For each CHF image, we reshape the image to 224x224x3 RGB for integration to VGG19. This will give us the high and low frequencies where the high values are assigned to the relevant features and the low values are the general background features [11] [10] [8]. The chain of images to construct CHF images is displayed in Fig. 2 for your reference.

3.2 Dataset Creation

We performed the following method for compiling a dataset of ASL and ISL number images zero through nine using both the pre-existing collections ASL and ISL, as well as our pre-processed images [6] [5]. Each dataset is a python3 numpy collection of objects with keys 'shaped_img' (mapped to the image of interest 224x224x3 expected for VGG19 with pretrained ImageNet weights[16]), 'language' (mapped to either the string 'asl' or 'isl'), and 'semantic' (mapped to the string of a number '0' through '9'). The ISL semantic datasets are comprised from the 12,000 original images on Kaggle's "Indian Sign Language Dataset" [5].

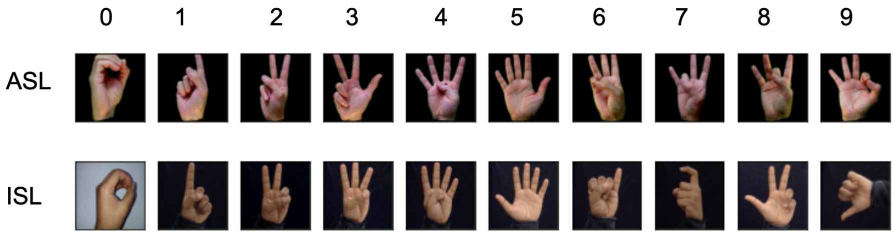

Fig. 1. Set of number system (0-9) in American Sign Language(ASL) and Indian Sign Language (ISL) [6] [5].

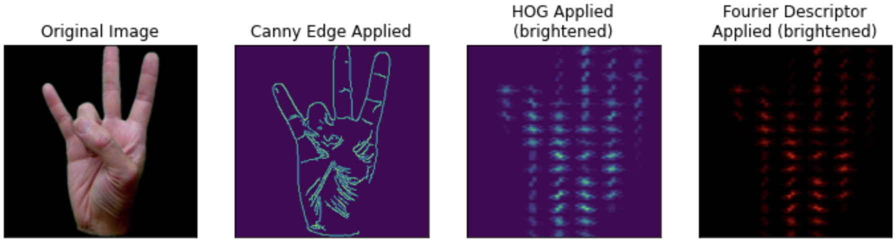

Fig. 2. (a) Original image, (b) Original Image with Canny-Edge Detection applied, (c) Original Image with Canny-Edge Detection and Histogram Oriented Gradients (HOG) applied, (d) Original Image with Canny-Edge Detection, HOG, and Fourier Descriptor applied (CHF).

The ASL semantic dataset is comprised from the 700 original images on Kaggle's "American Sign Language Dataset [6]. Finally, a language classification dataset is constructed as a composite of the ISL and ASL language datasets. We then repeat the process of ISL semantic, ASL semantic, and ISL/ASL language dataset with the CHF pre-processed images instead of original images. We recognize that the ISL/ASL combined dataset has a seventeen to one imbalance split of data which we can improve on in future work. These individual datasets are used to experiment in creating an optimal accuracy for our final architecture.

In or initial research, we presumed that this pre-processing would yield better results in language and semantic classification as this identifies relevant substructures of the human hand. After experimentation we realized a need to review the original images to establish a baseline for the experiment. This prompted the creation of the original images datasets. It was later found to have near perfect performance yielding our initial experiment design invalid.

3.3 Training

For training, we further decompose the above datasets into training, validation and test. We perform a 60/35/15 training, validation, and test split of the data for semantic classification to maximize the amount of training and validation data available to us. Due to RAM constraints in training systems during language classification, 20% of the data was removed from the ISL/ASL combined collection and a 48/28/12 training, validation and test split was used for language classification. The final dataset splits used for our experiments have been published for public use on Kaggle as "**asl_isl_numbers_conversions**" [17].

We experiment with four training types, namely: fine-tuning VGG19 on CHF images, transfer learning on CHF images, fine-tuning on original images, and transfer learning on original images. We use VGG19 as the base model for each experiment. We run experiments for transfer learning compared to fine-tuning to understand the impact on learning and training under these conditions.

For fine tuning, we allow all layers of the VGG19 with new softmax layer to learn. During fine tuning, we first trained on the ISL semantics, as it had a larger number of sample images, and then resulting features were used as a base network to fine tune over ASL dataset. Finally, those final features trained over ISL semantic, then ASL semantic, were used to train the ISL/ASL language classifier. In future experiments we recommend instead fine tuning on ISL/ASL language classification first, then ISL semantic, and then ASL semantic as this will front load larger datasets for fine tuning in the pipeline.

For transfer learning, we only allow the new softmax layer to learn, and we freeze the learning on the rest of the VGG19 network using a fresh VGG19 network with ImageNet weights for each classifier trained. We explore original images versus CHF images to understand the value of CHF pre-processing. For each experiment, we train with following constants to: adam optimizer, learning rate of 0.001, max epochs 50, and early stopping on condition that validation loss fails to decrease by 0.05% within two epochs. We chose these constants based on our experience with the data and known standard practices.

3.4 Architecture Design

Using the best performing models for each classifier, we constructed a translation architecture comprised of three classifiers: one VGG19 based Language Classifier (LC), one VGG19 based ASL semantic classifier (ASLC), and one VGG19 based ISL semantic classifier (ISLC). The model expects a 224x224x3 image as input. The architecture will pass this image to LC to determine the original language of the image. Once the original language is determined, the same image will be passed to the relevant semantic classifier, either ASLC or ISLC. The relevant image semantic is then determined by its respective language classifier. Finally, the system sources a relevant image from the opposite language and returns it to the user as demonstrated in Fig. 3. A sample of this architecture has been published on github for public use called "**EightOrThree**" [18].

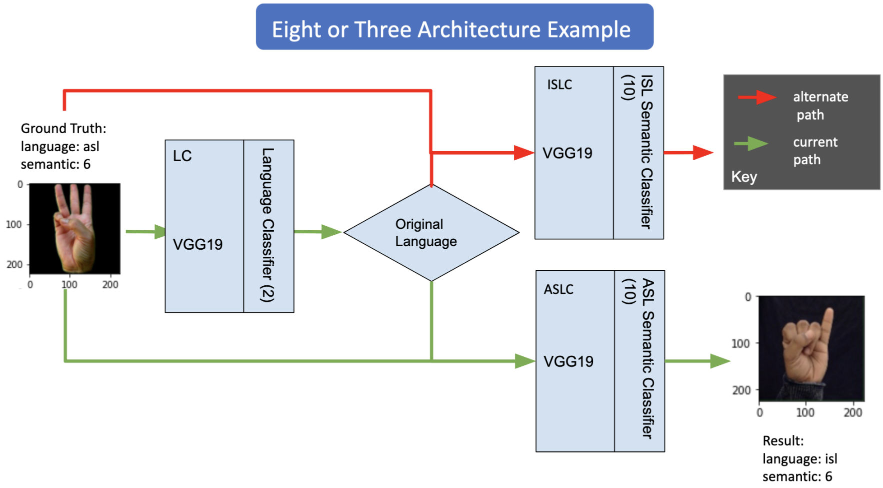

Fig. 3. Demonstrated above, following the green arrows, the ASL sign for 6 is passed to LC. This is successfully identified as ASL. The same image is then passed to ASLC and a prespecified image of the ISL sign for 6 is presented to the user of the tool. We note, that if an ISL image was presented to the tool instead, then the image would have been passed to ISLC following the red path out of "original language" decision instead of the green path.

4 Experimental Design

4.1 Accuracy

In order to measure the effectiveness of our model at sign language recognition and translation, we capture the training accuracy and validation accuracy at each epoch during training for each of the four training experiments. In addition, we capture the test accuracy of the best models for each experiment. This test accuracy metric is used to determine the best model amongst our different approaches, as this shows the viability of the translation architecture to work in the real world. By having high accuracy, we assure our end user that the tool can reasonably translated from one sign language to another.

4.2 Timing

The speed of our translation architecture to take in an image, identify its domain language, classify its semantic, and translate it to other language will be measured on the test data. Here, real time CHF pre-processing was not required for this architecture as we determine the best component models worked on original images. We select a threshold for processing time as 0.5 seconds as this represents a reasonable wait time for translation. If the speed of the model to perform translation is less than 0.5 seconds, then this demonstrates the proof of concept for real world applications and future robotic or virtual sign language translation assistants in real time.

5 Experimental Results

5.1 Accuracy by Epochs for ASL

As shown in Fig 4, of the four experiments ran, we found the least epochs to convergence on ASL semantic training while Fine Tuning the pre-possessed CHF Images converging at 6 epochs (patients to 8 epochs). We expect this is low because we first fine tuned the model in that experiment over ISL images prior to fine tuning on ASL, thus the base system already has the knowledge of CHF features. By contrast, transfer learning on CHF images took the most epochs (converges at 12 epoch with patients to 14 epochs). This is reasonable as the training dataset has only 420 images and the CHF images have contrasting features to those found in ImageNet. It is reasonable to expect more epochs to achieve the a comparable accuracy with less pre-training.

Also demonstrated in Fig. 4, the validation accuracy was observed to be higher than the training accuracy more consistently for the fine tuned experiments than the transfer learning experiments. This discrepancy could be because the hidden layers of the model in fine tuning were trainable so the gradient descent was applied on initial layers before validation accuracy is calculated. By contrast, in transfer learning, only the soft max layer is trainable and less learned parameters are changed between the calculation of training accuracy and validation accuracy.

5.2 Total Accuracy

Let us analyze the validation loss and validation accuracy of our models as displayed in Fig. 5. In each case we observe the highest value for accuracy on the

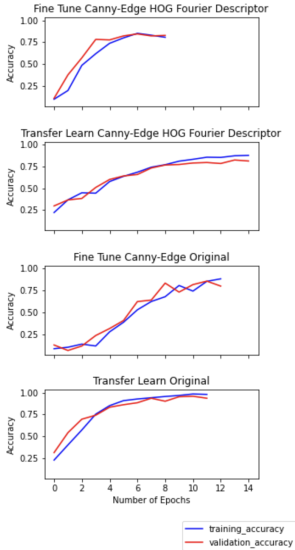

Fig. 4. Graphs of number of epochs compared to training accuracy (blue) and validation accuracy (red) measured in percentages - scaled zero to one - observed while training the ASL semantic classifier.

original images transfer learned on VGG19 with ImageNet weights including ISL semantic (100%), ASL semantic (95%), and the language classifier (100%). This demonstrates that VGG19 pre-trained on ImageNet contains all the knowledge necessary to perform image classification on ASL and ISL signed numbers. In the future research, we will explore baseline implementations before considering pre-processing like CHF.

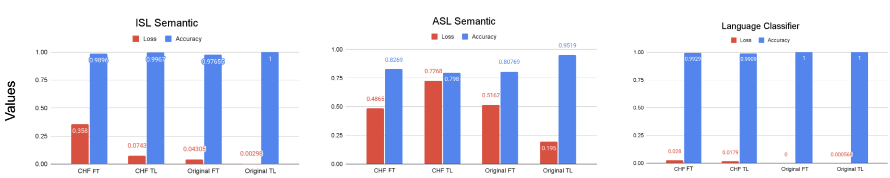

Fig. 5. Accuracy across different semantics using CHF dataset with fine tuning (CHF FT), CHF with transfer learning (CHF TL), original dataset with fine tuning (FT), and original dataset with transfer learning (TL), (a) ISL Semantic (b) ASL Semantic (c) Language Classifier

We note that the lowest accuracy of the component machines of our final architecture is the ASLC with accuracy of 95.2%. Noting this, our system is better able to translate from ISL to ASL with perfect accuracy on similar images to the original datasets, however, on ASL to ISL translation, there is the potential for some error. We believe this lower accuracy is due in part due to lower amount of images in the ASL dataset. In future experiments we would explore data generation and sourcing additional relevant datasets as detailed further below.

In language classification it was observed that the original dataset is outperforming the CHF fine tuning, this maybe due to the imbalance in the dataset ratio against the ISL versus ASL (seventeen ISL images per one ASL image). This can be rectified in the future by using higher quantity of quality images of ASL which would in turn increase the accuracy of the ASL semantics.

We note also, that our original concern about ISL eight versus ASL three was not captured in the original datasets. This was discovered after experimentation, but the original datasets had mirroring issues where one dataset would appear right handed and the other left handed. this is not a natural semantic of the sign languages and would not represent real human experiences like a signer with only one hand available. In doing more data generation, we could include mirror images of the ASL data to better simulate the real world. With that, we would expect issues in the language classifier, however, as we scale the research from numbers up to sentence level, we expect that sentence context in series data will help the language classifier understand the true language being put into the system.

5.3 Timing

Average timing across the 1,550 test training images were tested in the final architecture found to be 0.154 seconds to translate on average when running on GoogleColab GPU. This suggests the ability of this tool to be used in the real world for near real-time sign language translation. As the tool expands to heavier models like BERT for sentence translation, we expect that the time for translation will increase. Sentence level translation time will also be impacted by any need to have block input sequences rather than streaming data. We note that the initial thought to perform live pre-processing of the data was not necessary as the original ImageNet VGG19 had the best performance. If the sentence level translation needs real time pre-processing, that will also increase the time of computation for translation.

5.4 Future Work

The next steps would be to perform similar experiments at the sentence level. To support this, we'll need to first construct better datasets for non-ASL sign languages comparable in quality to the Boston University ASLLRP Continuous Signing Corpora [19]. At the sentence level, the proposed CHF pre-processing might want to be reconsidered as related work suggests these pre-processing steps can lead to performance gains. The final goal would be to create a live feed capturing sign language translator, for any language possible. We also note that our hardware could not train on a full 12,595 image dataset against VGG19 before running out of RAM, so better hardware would be required to train on videos and achieve comparable accuracy in a reasonable amount of training time.

6 Conclusions

We created a system to translate from ISL to ASL over numbers with perfect accuracy and ASL to ISL with 95% accuracy. This demonstrates a proof of concept for robotic or virtual assistant for real time sign language translation. We speculate that with better datasets we can achieve higher translation modalities, such as word or sentence level translation between the 300 sign languages of the world.

This work gives agency to the deaf and hard of hearing community during international travel, allowing them to better communicate within their communities across sign language barriers. This work also presents an opportunity to help young children to learn communication earlier as the motor skills necessary for sign language develop earlier than the vocal skills for human speech.

References

1. Kozik, K.: Without sign language, deaf people are not equal. Human Rights Watch (2019)
2. Anonymous: What are the different types of sign language? SignSolutions (2021)
3. Anonymous: What is american sign language (asl)? (unkown) Accessed 2022-11-30.
4. Adeyanju, I., Bello, O., Adegboye, M.: Machine learning methods for sign language recognition: A critical review and analysis. Intelligent Systems with Applications **12** (2021) 200056
5. Anonymous: Indian sign language dataset (2022) Accessed 2022-11-30.
6. Anonymous: American sign language dataset (2019) Accessed 2022-11-30.
7. Sujanani, A., Pai, S., Udaykumar, A., Bharath, V., Prasad, V.R.B.: Unidirectional ensemble recognition and translation of phrasal sign language from asl to isl. In Senjyu, T., Mahalle, P.N., Perumal, T., Joshi, A., eds.: Information and Communication Technology for Intelligent Systems, Singapore, Springer Singapore (2021) 241–249
8. Routray, S., Ray, A.K., Mishra, C.: Analysis of various image feature extraction methods against noisy image: Sift, surf and hog. In: 2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT). (2017) 1–5
9. Sharma, A., Mittal, A., Singh, S., Awatramani, V.: Hand gesture recognition using image processing and feature extraction techniques. Procedia Computer Science **173** (2020) 181–190 International Conference on Smart Sustainable Intelligent Computing and Applications under ICITETM2020.
10. Adeyanju, I., Bello, O., Azeez, A.: Development of an american sign language recognition system using canny edge and histogram of oriented gradient. Nigerian Journal of Technological Development **19** (09 2022) 195–205
11. Kishore, P.V.V., Prasad, M.V.D., Prasad, C.R., Rahul, R.: 4-camera model for sign language recognition using elliptical fourier descriptors and ann. In: 2015 International Conference on Signal Processing and Communication Engineering Systems. (2015) 34–38
12. Starner, T., Pentland, A.: Real-time american sign language recognition from video using hidden markov models. In: Proceedings of International Symposium on Computer Vision - ISCV. (1995) 265–270
13. Premjith, B., Kumar, M.A., Soman, K.: Neural machine translation system for english to indian language translation using mtil parallel corpus. Journal of Intelligent Systems **28**(3) (2019) 387–398
14. Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., Gao, J.: Deep learning–based text classification: A comprehensive review. ACM Comput. Surv. **54**(3) (apr 2021)
15. Khurana, D., Koli, A., Khatter, K., Singh, S.: Natural language processing: state of the art, current trends and challenges. Multimedia Tools and Applications (2022)
16. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition (2014)
17. Wilson, S., Yadav, A.: asl_isl_numbers_conversions Accessed: 2022-12-11.
18. Wilson, S., Yadav, A.: Eight or three (2022) Accessed: 2022-12-08.
19. University, B.: Asllrp continuous signing corpora Accessed: 2022-12-12.