ASL Hand Sign Classification and Prediction

Arden Chew[1], Amama Mahmood[1], Zach Silver[1],
Aniruddha Tamhane[1], Wei-Lun Huang[1]
Johns Hopkins University, Baltimore-MD[1]

Abstract - **Sign language is a hand-gesture based means of communication for people suffering from auditory impairments. While there are already a variety of applications that help translate English to sign language, the inverse problem of translating sign language to English is an active area of research. In this project we have developed a novel method for translating a sequence of hand gestures into English words by ensembling a convolutional neural network (CNN) based image classifier with an n-gram letter predictor. We have employed data augmentation techniques for a robust model training. We have implemented our model on the MNIST sign language dataset[1] yielding a prediction accuracy of 98.15%.**

BACKGROUND

Current research on sign language translation implements a variety of statistical computer vision (Histogram of Gradients), machine learning (SVM), and Neural Network (CNN) based classification techniques; however, these techniques fall short of achieving pragmatic accuracy and confidence.

Classical orientation histogram methods for gesture based recognition have achieved sufficient usability in dynamic human-computer interface control systems, yet this usability remains limited to small subsets of gestures, often sets of five or fewer (stop, left, right, up, down)[2]. It has been noted that success with these models requires that the controlling hand dominate the image frame, and that the hand's orientation remain relatively constant[2]. Sign language, however, requires the usage of multiple hands and varying hand orientations, and can even derive meaning from changes in body posture. Furthermore, modern machine learning and neural network based approaches experience similar training data-based shortcomings. Despite validation success, low testing accuracies have been attributed to left vs. right hand shifting, gender of signer, skin tone, rotation, orientation, and size[2]. To address these issues in a succinct way, we have augmented our training dataset by way of hand contour extraction from the initial MNIST sign language dataset, followed by an allotment of transformations that build a predictor robust to the aforementioned variations.

In a more recent study (2014) conducted by the National Taipei University of Technology, a gaussian mixture model was first used to alleviate variance in hand types, before being fed into a standard eight-layer feed-forward neural network[3]. This improved methodology was instrumental in achieving the modern state of the art average gesture recognition rate of 95.56%[3]. Yet a greater than 4% error rate can still lead to significant errors in meaning derivation, particularly in word formation. It is evident that meaning derivation in the sign language alphabet comes from both local and extrapolated image region features. We have therefore implemented residual learning layers in our neural network architecture. Furthermore, natural language processing offers a method for letter based predictions independent of the hand sign itself. Through ensemble methods balancing our residual neural network model with our natural language processing model we are able to improve upon the accuracy of our pure deep learning method.

PROPOSED METHOD

We aimed to improve upon state of the art models by diversifying our training dataset via modern data augmentation techniques, refining the CNN architecture choices, while ensembling with predictive natural language processing techniques.

A. Data Augmentation

As noted, a diversified training dataset was necessary for our model to generalize well in scenarios of differing gesture orientation, abnormal relative frame-hand location, and variation in inherent hand characteristics. To build this dataset, we concentrated efforts on a robust hand extraction method, then employed a probabilistic suite of transformations to the extracted hand, before superimposing the product image on a generated background.

Our method of hand extraction was two fold. Firstly, an experimentally determined threshold was determined to predict the presence of skin in an image. Due to the nature of skin tone, images were thresholded in HSV format. Secondly, a pre-trained Haar-cascade classifier was used to generate potential hand contour candidates. Finally, the candidate with maximum skin colored pixels was selected.

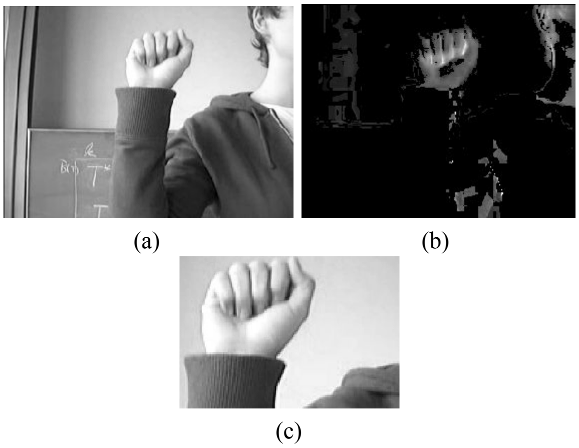
(a)
(b)
(c)

Fig 1. Grayscaled progression of letter 'a' gesture extraction: (a) raw image (b) intensity mapping from skin tone thresholding (c) haar cascade motivated cropping

This extraction method was conducted over a set of 240 labeled training images. The following data augmentations measures were used on the cropped images to build diversified training and validation datasets spanning 10,642 images and associated labels: random rotation bounding rotations to ±180°; random crop, maximum cropping to 0.9 height and width; flips about a vertical axis of symmetry; Gaussian noise (white) addition, variance $\sigma^2 = 0.25$, bounding absolute to ±1.0 to simulate lighting conditions; modification of image contrast; random shearing; randomized pixel salting probability with $p = 0.05$; and randomized translations. The random angle of rotation, random translation lengths and random shear coefficients were drawn from a truncated normal distribution having a zero mean and variances equal to 30º, 20 pix and 0.5 respectively.

Further, each augmentation technique was applied to an image with a random probability of 0.3. Thus, a dataset of 10,642 images was generated out of a possible of 61440 ($240x2^8$) images.

B. Deep Learning

We experimented with a series of network architectures including modified U-Net variations, feed-forward CNNs, and a simplified Alexnet implementation, before settling on a deep residual network.

Our best performing network included a 26 layer residual network with dropout. Residual layers offer the added advantage of incorporating both local and distant image features into each layer's input, which is particularly necessary in gesture recognition. Each finger's relation to other fingers can play an equally defining role as the as the role of the individual finger's orientation in creating meaning. In addition, skip layers in our residual network mitigate the recurring vanishing gradient issue, identified in previous gesture classification studies[3].

Each residual layer includes a series of 3x3 convolutional layers downsampling to a two-fold reduction in both height and width. The architecture is initialized with a 7x7 kernel convolutional layer, followed by batch normalization, and ReLU activation. It is then followed by 24 residual layers which incorporate both a 2D dropout layer integrated in the convolutional pass, in addition to a linear dropout layer. Dropout is conducted at a probability of $p = 0.05$ but similar results are achieved with $p = 0.03$. The final convolutional layer is average pooled via adaptive average pooling before entering a linear layer that reduces to the classes associated with each sign. A softmax transformation is used to identify the associated probabilities of association between an input image and each gesture. In addition, adam optimization was utilized with its default parameters as it provides superior generalization to stochastic gradient descent. We believe this to be a result of the common local minimas found in gesture detection on account of similarities amongst some letters.

Hyper-parameter tuning found the optimal set to be as follows: 20 epochs, 0.001 learning rate (Adam optimizer), 0.5 dropout probability and 8 batch size (note that this is subject to change given the probabilistic nature of this model). Batches were parallelized and computed using the Tesla K80 GPU hosted on Google Colaboratory.

C. Natural Language Processing

The natural language processing portion of our model depends on determining the probabilities of arbitrary *n*-grams in English. An *n*-gram is a group of *n* letters. For example, in order to determine the most common 2-letter combinations (bigrams), we looped through thousands of English texts and counted every bigram in a 24 by 24 array (with the letters *j* and *z* missing). The same method was used for trigrams and 4-grams, using 3 and 4 dimensional arrays respectively. The dataset we used, the Brown Corpus, is a collection of texts from hundreds of sources, so it should provide an accurate picture of English *n*-gram frequencies.

After all the *n*-gram probabilities were calculated, we used them to augment the results of our Deep Learning model. When predicting a letter, we take into account both the output of our deep network and the most probable outputs given previous letters. For example, if our word currently consists of the letters "t" and "h," the NLP model would make it much more likely that the next letter predicted would be "e" or "a," since these are some of the most common three-letter combinations in English.

D. Ensemble Methods

While sign language gesture prediction requires success in letter-to-letter prediction, the language draws meaning from a series of gestures which complete words. Ensembling both our deep learning classification model with our natural language predictive model provided for enhanced accuracy in classifying individual letters on the basis of context. As can be seen in our results, despite the success of our residual layered deep network, the accuracy for prediction at word level can drop dramatically, especially for words containing letters that are ambiguous to CNN classification. Ensembling with our n-gram predictor provided contextual information within a word, which improved accuracy across all tested data. It is necessary to note that testing data for the ensemble method was individually generated so as to create word groupings.

Our method for ensembling was simply a probabilistic product of our natural language processing and deep learning models. The natural language model outputs a predictive tensor with the associated probability of each class being the next sign. Similarly, given the next sign in image form, our deep learning model provides a full tensor of probabilities for each gesture class. The ensembling method iteratively multiplies the probability from both models to determine the most probable next gesture.

RESULTS

Our model was successful in surpassing state of the art testing. Given our residual network alone, we were able to best achieve a generalized testing accuracy of 98.15% on the Kaggle provided MNIST sign language dataset[1]. The letter-by-letter accuracies are detailed in the confusion matrix provided by figure 2.

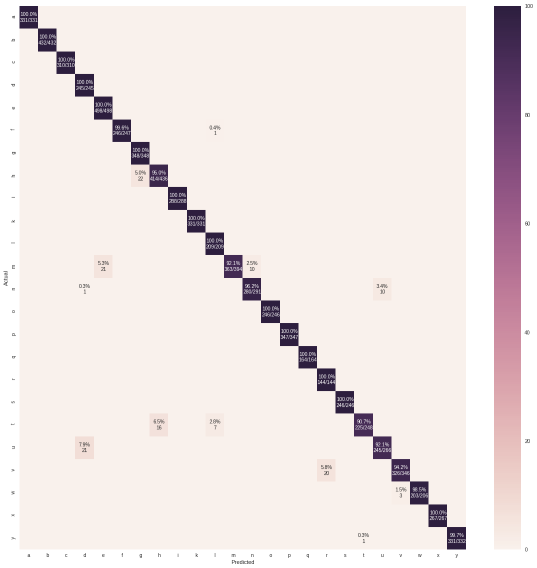

Fig 2. Confusion Matrix Validation Accuracy of Neural Network.

Furthermore, our ensembling method was successful in improving upon the success of our deep residual network. Given our testing data set (generated from the 1000 most common English words[4]), for fully signed words our generalized neural network model achieved a testing accuracy of 90.89% in correctly translating entire words. When ensembled with our natural language processing model, this accuracy improved to 93.89%. More specifically, while n-gram ensembling sparingly improved short word, 4 or fewer letters, accuracy from 94.15% accuracy to 94.56% accuracy, it significantly improved medium word, 5-8 letters, and long word, 9+ word, accuracies from 89.95% to 93.88% and from 83.04% to 92.30% accuracy respectively. Our methods are simple and fast. The residual network is of appropriate depth to achieve real time translation, and when paired with a sufficient GPU, our natural language processing techniques can be pragmatically integrated as well.

REFERENCES

[1] Tecperson. "Sign Language MNIST." Kaggle, www.kaggle.com/datamunge/sign-language-mnist.

[2] Zhou, Hanning, et al. "Static Hand Gesture Recognition Based on Local Orientation Histogram Feature Distribution Model." 2004 Conference on Computer Vision and Pattern Recognition Workshop, Dec. 1994.

[3] Lin, Hsien-I, et al. "Human Hand Gesture Recognition Using a Convolution Neural Network." 2014 IEEE International Conference on Automation Science and Engineering (CASE), 2014.

[4] "Google-10000-English" Github, www.github.com/first20hours/google-10000-english.