

ASL FINGERSPELLING RECOGNITION THROUGH HIDDEN MARKOV MODELS

MATTHEW SO

Advisor

Name

Signature

Second Reader

Name

Signature

TABLE OF CONTENTS

Chapter 1: Introduction	1
Chapter 2: Literature Review	3
Chapter 3: Methodology	7
3.1 Data Collection	7
3.2 Models	8
3.2.1 Uniletter Hidden Markov Models	9
3.3 Evaluation	10
3.3.1 PopSign Evaluation	10
Chapter 4: Results	12
Chapter 5: Discussion	16
Chapter 6: Conclusion	17
Chapter A: Word Lists	18
References	20

CHAPTER 1

INTRODUCTION

Like most sign languages, American Sign Language (ASL) is notable for its unique grammatical structures such as classifiers and its use in a more complex [1] multi-dimensional (spatial) medium, rather than a uni-dimensional (auditory) medium. These features significantly challenge ASL recognition and translation systems. As a result, most current recognition or translation systems require expensive equipment such as accelerometers and depth cameras, restrictions on recognizable phrases, and/or low recognition accuracy [2][3][4][5]. For example, the state-of-the-art commercial ASL recognition and translation system SignAll requires four RGB cameras, one depth camera, and custom colored gloves to accurately recognize basic ASL, leading to an unreasonably high cost (\$3300/year) [6]. These systems' inaccessibility effectively require adult learners, who require continuous practice, to affiliate with an institution or attend in-person or one-on-one sessions with instructors to learn generative (practicing ASL by producing signs) skills. These requirements' time consuming nature often deters potential learners from learning ASL. Because continuous communication between parent and child is vital to develop language skills, the resulting parental dissuasion from learning ASL can also prevent deaf children from communicating with hearing parents, preventing language acquisition during their critical period of development. This deficit leads to language deprivation syndrome, which impairs working memory [7] and leads to poor life outcomes [8][9][10][11]. However, ASL fingerspelling (signs corresponding to the letters of the alphabet), a subset of ASL, may be able to engage and encourage parents for initial learning, and an automatic tutor may be more feasible using computer recognition with off-the-shelf hardware.

While the complex hand shapes forming each sign pose a challenge for hand pose estimation, which is vital for sign recognition, fingerspelling has a limited number of signs and no classifiers, potentially making it less dependent on complex equipment for recognition. Furthermore, as a

more accessible means of instruction, generative ASL fingerspelling games also provide opportunities to introduce ASL in a limited form to learners, which can encourage further ASL learning through other traditional means. While laptops, depth cameras, wearables, and other systems have been shown to be capable of accurately recognizing ASL fingerspelling [12][13][14], such devices are not generally widely available or convenient, limiting broader usage of fingerspelling recognition systems. Thus, in this project, we have designed, tested, and analyzed recognition systems capable of running on PCs and mobile devices.

To create recognition models, we have analyzed data sufficiency by collecting training data from three users on 36 fingerspelled words of varying lengths. With this data, we have analyzed the performance of recognition models, specifically uniletter and triletter Gaussian Mixture-Hidden Markov Models (GMM-HMMs), on various datasets to determine the viability of fingerspelling recognition on desktop and mobile environments. Furthermore, we are porting the game to Pop-Sign, a mobile game teaching adults to sign, and have determined the best possible implementation of fingerspelling recognition into the game by determining the ideal sets of words to be recognized in the game. Through our efforts, we hope to broaden the reach of ASL-based instruction to the wider public and improve the skills of adult learners and the deaf children indirectly affected.

CHAPTER 2

LITERATURE REVIEW

ASL's distinct challenges have limited the reach of ASL recognition and translation systems by requiring inaccessible equipment such as accelerometers and depth cameras, restrictions on recognizable phrases, and/or low recognition accuracy [2][3][4][5], limiting the reach and effectiveness of generative (practicing ASL by producing signs) instruction for learners. However, ASL fingerspelling is significantly simpler to differentiate than ASL, which face significant challenges to recognition systems, as will be shown. Given these factors, the use of fingerspelling increases the flexibility of deployment platforms and is thus worth investigating. As part of this, numerous design elements, including game design, will be needed, with two primary parts warranting attention: robust hand pose estimation, and user-independent language recognition and verification.

Hand pose estimation has been variously approached with a variety of methods. One of the earliest approaches, as exemplified by Dornet et al. [15] and Dong et al. [16], used colored gloves to identify joints of the hand. Using such equipment allowed each joint to be individually identified by simply identifying the color of regions, for which existing, highly accurate, computer vision methods exist. While this approach is simpler, and the results produced have been shown to be effective with simple models such as Random Forests (RF), the need for gloves limits its broader applicability. In a similar vein, Pansare et al. [17] used edge orientation to identify hand poses. While this technique avoided the need for custom gloves, its lack of testing on more than one user, along with relatively poor results (under the best conditions, 88% user-dependent accuracy was achieved), likewise rules this technique out for practical use.

The rise of deep learning techniques has led to a re-examining of this problem. Ge et al. [18] outlines one such state-of-the-art method. They showed that shallow convolutional neural networks (CNN) were able to robustly, accurately, and efficiently perform 3D hand pose estimation, which would be necessary to recognize fingerspelling. The performance of their model (215 fps using

only one GPU) suggests that hand pose estimation is not only practical but fully feasible on a mobile device, which this project aims to develop on. In a similar vein is Koller et al's work [19] on the recognition of a million hand images. The use of expectation maximization and CNNs created a robust hand shape classifier capable of recognizing moving, as opposed to static, hand shapes with a top-1 training accuracy of 55.3% on the RWTH-PHOENIX-Weather-2014 dataset, which contains containing low-resolution signing of weather forecasts. Importantly, their work trained hand shape identification using noisy sequence labels, as opposed to manual labelling of hand shapes. While this method was not tested on more complex datasets (the signer in the testing dataset is on a single-colored background), their training method dramatically increased the amount of usable training data, thus improving accuracy.

User-independent language recognition and verification is also necessary to process user signs and thereby affect the output of the game. Camgoz et al. [20] notably does so with the Transformer architecture. By using Connectionist Temporal Classification (CTC) loss, which binds the recognition and translation problems associated with video segmentation into signs, ground-truth timing information was rendered redundant, leading to lower error rates in translation of the RWTH-PHOENIX-Weather-2014 dataset by using non-timed data. While many of the flaws from Koller et al.'s method remain, the broader techniques outlined could broaden this project's usable data.

The rise of existing wearable devices such as smartwatches has also led to novel methods for fingerspelling recognition such as, most notably, Hou et al.'s work [21] on a smartwatch-based sign translator. Using a CTC system on LSTMs, a smartwatch equipped with an accelerometer, when tested on five users and approximately 100 fingerspelling hand poses and signs, was able to reliably recognize signs. This system achieved a user-dependent and user-independent new-user word error rate of 1.04% and 10.7% on sentences, respectively. Note that models with high user independence ensures greater user friendliness and versatility in the final product, as no supervised training period requiring existing knowledge of ASL fingerspelling would be needed for players to begin the game. Thus, although smartwatch (or more generally, accelerometer-based) systems have become more prevalent, their low user-independent accuracy rules out use in this project.

In addition to hand pose estimation, studies have been performed to determine the effects of language-based games on child development. The most notable of these is CopyCat, developed by Weaver et al. [22]. CopyCat aimed to improve the language skills of children through an interactive game, where the player would describe through sign a scene in the game in order to practice generative (the ability to produce language) skills. In contrast to our system, however, their game recognized ASL, as opposed to fingerspelling. To quantify the effects of the game on language skills for children, they performed a study at a local school for the deaf on 12 participants, aged between 6 and 11, with six control and six game participants. They were tested with a Wizard-of-Oz configuration of the game, whereby a human manually identifying signing substitutes for a computational recognition system. Ultimately, the study noted that children playing CopyCat experienced significant improvements in receptive, expressive, and sentence repetition abilities. While fingerspelling may yield differences in results due to fingerspelling's manual spelling of words, their work shows the positive effects of language-based games on language development for children, thus broadening the target audience for a fingerspelling-based game.

Though hand pose estimation holds importance, other components, including phrase verification, are necessary. In fact, Zafrulla et al. [23] and Yin et al. [24] demonstrated a potential solution. By using Hidden Markov Models (HMMs), which are generally well-suited towards language pattern recognition, and variations such as Segmentally Boosted HMMs (SBHMMs), they designed a verification algorithm with accuracy far greater than a standard classification algorithm. Specifically, the log-likelihood value of a forced alignment of a trained Hidden Markov Model, if treated as an accuracy rating, can be used to verify the accuracy of signed phrases using a threshold calculated based on existing log-likelihood values. This method extracted 82% accuracy from a 67% accuracy model trained using 1204 data samples collected from 11 deaf children [23]. Since the project requires accurate recognition and verification systems in order to achieve its aims, and the models shown above required comparatively low amounts of data to train due to using Hidden Markov Models, which offer greater simplicity than more complex neural networks, their work shows the continued potential of HMMs as underlying models for this project.

The works cited above collectively indicate both the existing difficulties with recognizing ASL with a non-static camera and the possibility of recognizing fingerspelling on mobile devices. Using hand pose estimators, which have been shown to be efficient and robust, recognition systems, and recognition systems, which have been shown to achieve high accuracies, it is also evident that a fingerspelling recognition system on mobile devices is genuinely viable. Thus, with data collected from mobile phones, recognition models such as HMMs and Transformers, and modern hand pose detection systems, an ASL-based instructional game has the potential to reach a far wider audience than a standard ASL game. Most importantly, such a system would benefit adults both through learning fingerspelling and through encouraging learning through other forms, thereby benefiting deaf children through communication with hearing parents.

CHAPTER 3

METHODOLOGY

This paper explores the feasibility of fingerspelling recognition on mobile devices by examining the performance of fingerspelling recognition model structures on various datasets. We analyze this through collecting data from users and testing model performance on tied-state triletter hidden markov models (HMMs). Furthermore, this paper identifies five-word partitions of the 36-word dataset with the lowest recognition errors in each partition by applying the highest accuracy model type onto a significant subset of potential five-word partitions in order to identify the optimal sets of words to use in PopSign, a mobile game teaching sign language, which uses fingerspelling models.

3.1 Data Collection

To test our output, we collected data containing twelve sets of a 36-word list (in the Appendix, Figure A.1) from three users using the Azure Kinect, a high-resolution RGBD camera, at 2160p resolution. The Azure Kinect was chosen due to its high resolution and ease of manipulation, since Kinect data can be augmented to form mobile-equivalent data. The following steps were taken:

1. Two users, one male and one female, who are both right-handed were selected in order to provide a range of hand shapes.
2. Approximately 12 sets consisting of 36 words each were recorded from each user. During recording, the order in which words were presented was randomized for each set to reduce the effects of sequence memorization on fingerspelling.
3. The collected videos were converted to arm and hand shape features using the latest version of the Mediapipe pose estimator (v.0.8.7.3) to convert videos into a form usable by recognition models. Since fingerspelling only involves the hand shape and relative motion of the hand, the only features (data points) used are the right hand landmarks.

This data is used to train and test various recognition models.

3.2 Models

For our experimentation, one type of models capable of recognizing sign language were tested: tied-state triletter HMMs. These models are generated using HTK, a toolkit used to train and test HMMs. For reference, we will describe uniletter HMMs to better describe tied-state triletter HMMs.

For each word, Hidden Markov Models are trained by performing a flat initialization, then performing iterations of Baum-Welch re-estimation (Algorithm 1) to refine the model. Every 20

Algorithm 1 Baum-Welch Re-estimation

- 1: $X = \text{Features}, Y = \text{Label}, T = \text{Total Time steps}$
 - 2: **Randomize HMM Parameters** $\theta = (A, B, \pi)$
 - 3: $\alpha(X_0) = P[Y_0, X_0] = P[Y_0|X_0]P[X_0]$
 - 4: $\beta(X_T) = 1$
 - 5: **for** $i = 0 \rightarrow \text{iterations}$ **do**
 - 6: **for** $k = 0 \rightarrow T$ **do**
 - 7: $\alpha(X_k) = \sum_{X_{k-1}} \alpha(X_{k-1})P(X_k|X_{k-1})P(Y_k|X_k)$
 - 8: **end for**
 - 9: **for** $k = N \rightarrow 0$ **do**
 - 10: $\beta(X_k) = \sum_{X_{k+1}} \beta(X_{k+1})P(X_{k+1}|X_k)P(Y_{k+1}|X_{k+1})$
 - 11: **end for**
 - 12: $\eta(X_k) = \frac{\alpha(X_k)\beta(X_k)}{\sum_{X_k} \alpha(X_k)\beta(X_k)}$
 - 13: $\epsilon(X_k, X_{k+1}) = \frac{\alpha(X_k)\beta(X_{k+1})P[X_{k+1}|X_k]P[Y_{k+1}|X_{k+1}]}{\sum_{X_k} \alpha(X_k)\beta(X_{k+1})P[X_{k+1}|X_k]P[Y_{k+1}|X_{k+1}]}$
 - 14: $\pi_0^* = \eta(X_0)$
 - 15: $A_{ij}^* = \frac{\sum_k \epsilon(X_k=j, X_{k-1}=i)}{\sum_k \eta(X_{k-1}=i)}$
 - 16: $B_{ij}^* = \frac{\sum_k \eta(X_k=i)1_{Y_k=j}}{\sum_k \eta(X_k=i)}$
 - 17: **end for**
-

mixtures, the number of mixtures in each HMM model is increased. Finally, Viterbi Decoding (Algorithm 2) decodes the HMM to find the sequence of HMMs with the highest observation likelihood. The number of states used varies based on the type of HMM trained and the data that will be used to train the HMM.

Algorithm 2 Viterbi Decoding

```
1: initialize  $viterbi[N, T]$ 
2: for  $i = 1 \rightarrow N$  do
3:    $viterbi[i, 1] = \pi_i * b_i(o_1)$ 
4:    $backp[i, 1] = 0$ 
5: end for
6: for  $j = 2 \rightarrow T$  do
7:   for  $i = 1 \rightarrow N$  do
8:      $viterbi[i, j] = \max viterbi[i', j - 1] a_{i',s} b_i(o_j)$ 
9:      $backp[i, j] = \operatorname{argmax} viterbi[i, j - 1] a_{i',s} b_i(o_j)$ 
10:  end for
11: end for
12:  $bestpropagation = \max viterbi[s, T]$ 
13:  $bestpathpointer = \operatorname{argmax} viterbi[s, T]$ 
14:  $bestpath = \text{path beginning at } bestpathpointer \text{ and continued using } backp \text{ data}$ 
```

3.2.1 Uniletter Hidden Markov Models

Uniletter HMMs assume that each letter of each word is fully independent from all other letters when signing. This approach is taken by CopyCat [25] due to the distinctiveness of signed words from each other and the short length of phrases. However, fingerspelling differs due to the prevalence of coarticulation, where multiple letters of a word are melded together. A uniletter HMM would be unable to consistently identify such letters and thereby the word due to lack of distinctiveness.

Triletter HMMs, in contrast, recognize words based on sequences of triletters. For example, the triletter sequence for *cats* is `sil0 c+a c-a+t a-t+s t-s sill`, where `sil0` marks the beginning of the word and `lill` marks the end of the word. By including context for each letter, potentially higher accuracy is possible. This is achieved by cloning the uniletter version and re-estimating by tying transition matrices using triletter transcriptions, as shown in Figure 3.1. Furthermore, tied-state triletter HMMs allow words not already in the dictionary to be recognized. Because of their extensibility towards new words not in the training set, only tied-state triletter HMMs are tested.

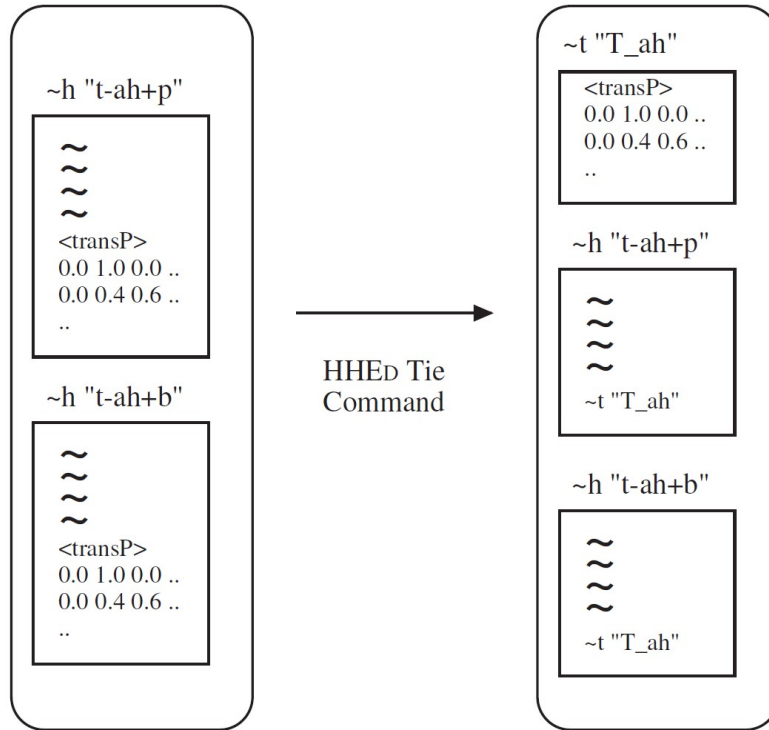


Figure 3.1: Transforming Uniletter HMMs to Triletter HMMs

3.3 Evaluation

The 36-word dataset (Figure A.1), which is generated from the MacArthur-Bates Inventory, a word list of the most important words for children to learn, will be used to evaluate the performance of the two model types. To do this, letter error and word error will be evaluated for each user using a test-on-train structure, where training data is used to test model accuracy, and an 8-fold cross-validation system, where eight trials are performed and in each trial, an eighth of the data is removed from the training set and is instead used to test the model. All training is performed on a standard desktop PC with an 8-core/16-thread CPU and 16 GB of RAM.

3.3.1 PopSign Evaluation

In addition to standard testing on the 36-word dataset, we will determine the effects of using subsets of the 36-word dataset by determining the performance of our HMM recognition models on a 15-word subset of the 36-word dataset comprising all three-letter words. In doing so, we will

determine the performance of our models on small datasets with similar types of words.

CHAPTER 4

RESULTS

The results highlight the particular importance of using relative, rather than absolute, joint position data. With non-centered hand features (raw finger and joint position data relative to the video frame), accuracy was relatively low, with error rates at 18.56% at best (User 2, Letter Error Rate) and 56.82% at worst (User 1, Word Error Rate) (Figure 4.1, Figure 4.2). This can be attributed to smaller size of hands when compared to the body. In whole-body signing, as is the case in CopyCat [25], the body occupies a large part of the frame of the video. As a result, body positions can be captured absolutely. In contrast, smaller hands mean that individual finger positions, although they may be meaningful when compared with other hand joint positions, are not meaningful on their own, since the location of the hand can significantly vary between sessions and videos. As a result, the use of centered hand features (hand joint positions relative to the wrist) dramatically improved the letter and word recognition accuracy of User 1 and User 2's signing to a level sufficient for educational use [23]. In contrast, wrist data did not significantly improve letter or word accuracy, and in fact worsened accuracy when delta wrist features (change in wrist position from frame to frame) was included (Figure 4.1, Figure 4.2). This is likely due to fingerspelling largely communicating information through finger positions. While wrist data, which stands in for whole-hand data, can be useful when identifying single and double letter combinations, the lack of minimal pair words which are only distinguishable by hand position limits its usefulness. Since the training and testing dataset used is small, the HMM model may have excessively emphasized wrist data, causing overfitting.

The performance of our models on the 15-word dataset was particularly surprising. As expected, because the 15-word dataset is smaller than the 36-word dataset, word and letter error rates were lower when using uncentered hand joint features (Figure 4.3, Figure 4.4). However, on centered data, the 15-word dataset ultimately performed worse. For example, the 15-word dataset

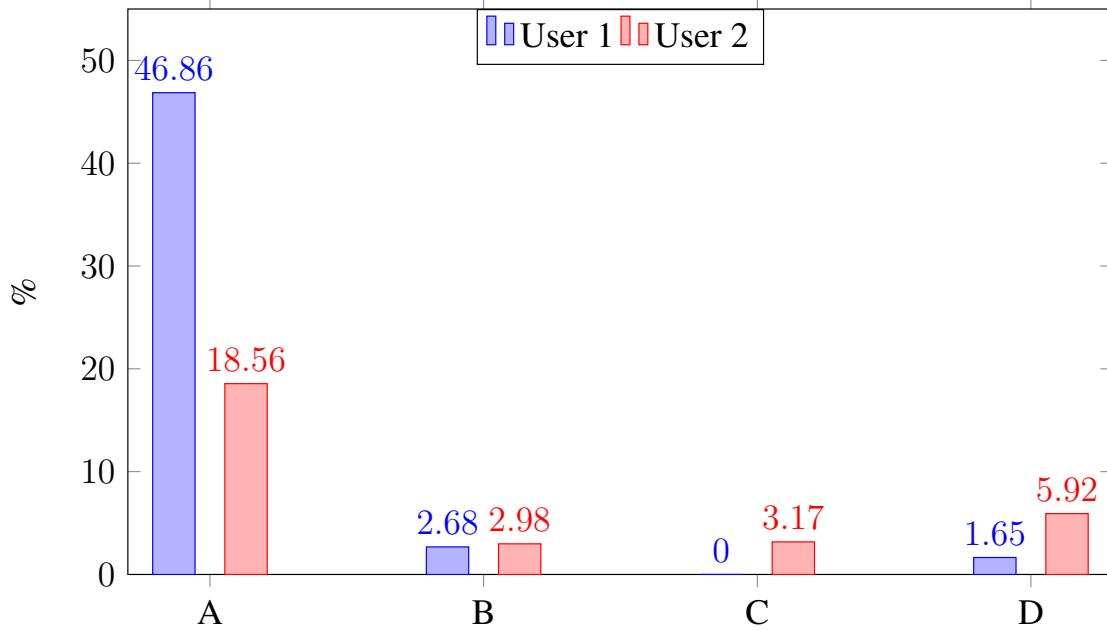


Figure 4.1: 36-word Letter Error Rate. Features used: A: Non-Centered Hands, B: Centered Hands, No Wrist, C: Centered Hands, Absolute Wrist, D: Centered Hands, Absolute and Delta Wrist

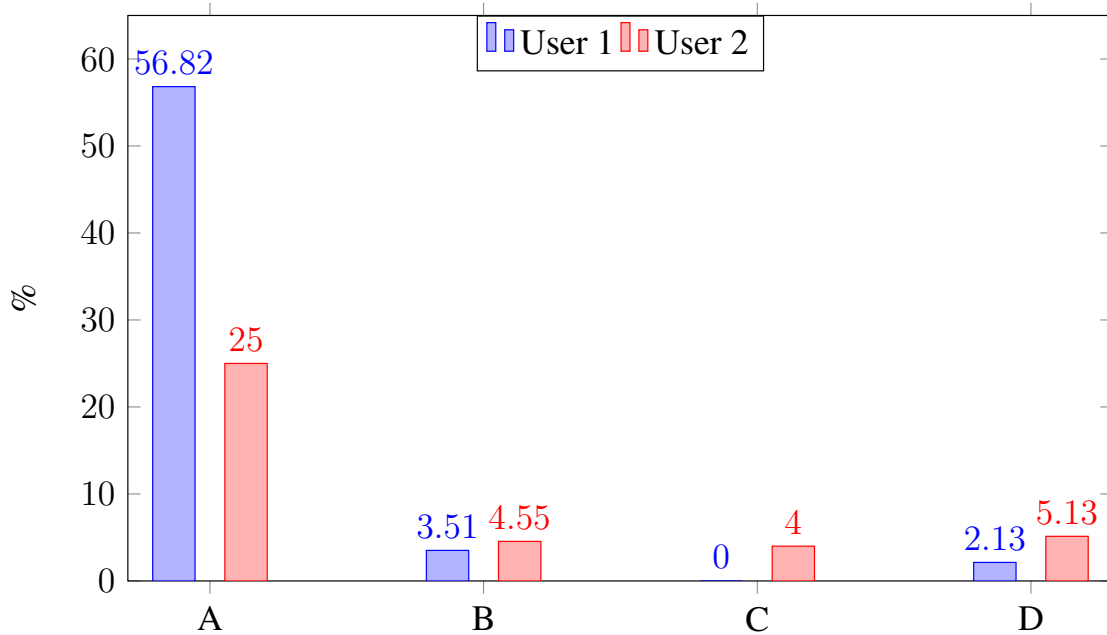


Figure 4.2: 36-word Word Error Rate. A: Non-Centered Hands, B: Centered Hands, No Wrist, C: Centered Hands, Absolute Wrist, D: Centered Hands, Absolute and Delta Wrist

achieved 3.58% and 8.45% word error rates (Figure 4.3), while the 36-word dataset achieved 3.51% and 4.55% word error rates (Figure 4.4) given the same features (centered hands, no wrist) for users

1 and 2, respectively. This is likely due to the paucity of available data. While additional words add complexity to the model, they also provide additional data for each letter, improving overall accuracy. Based on this, the proposed method of training individual models for small clusters of words to improve accuracy is likely not viable unless the cluster of words trained significantly differ in length.

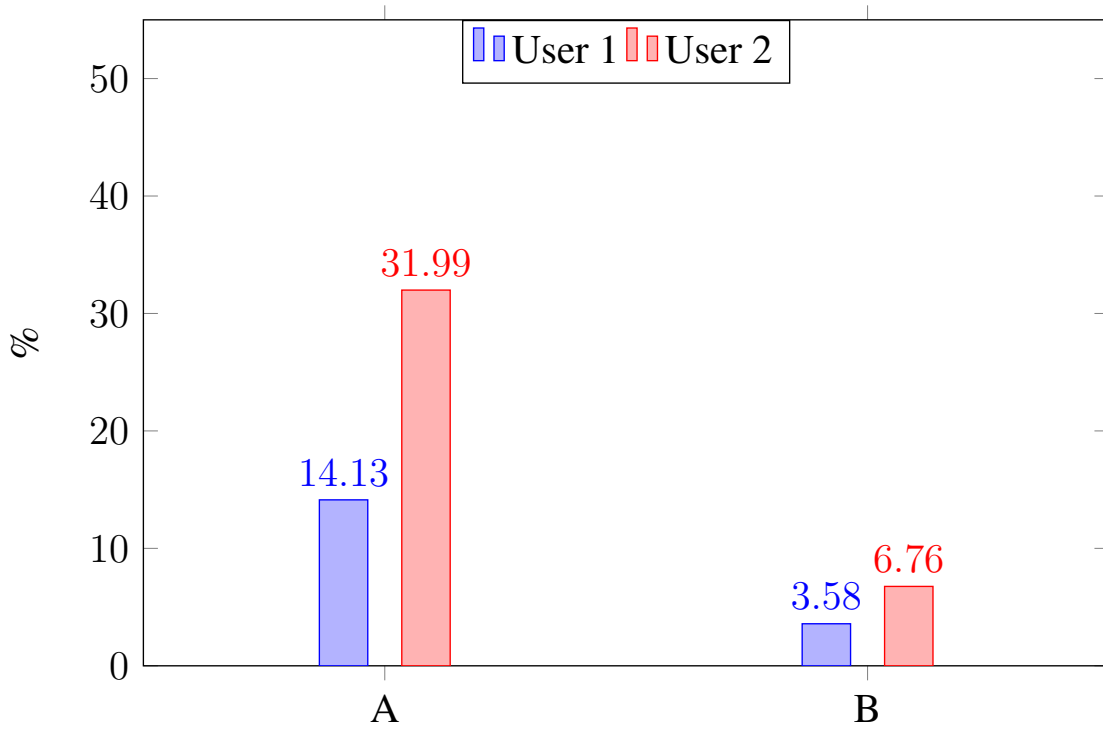


Figure 4.3: 15-word Letter Error Rates. Features used: A: Uncentered Hands, B: Centered Hands, No Wrist

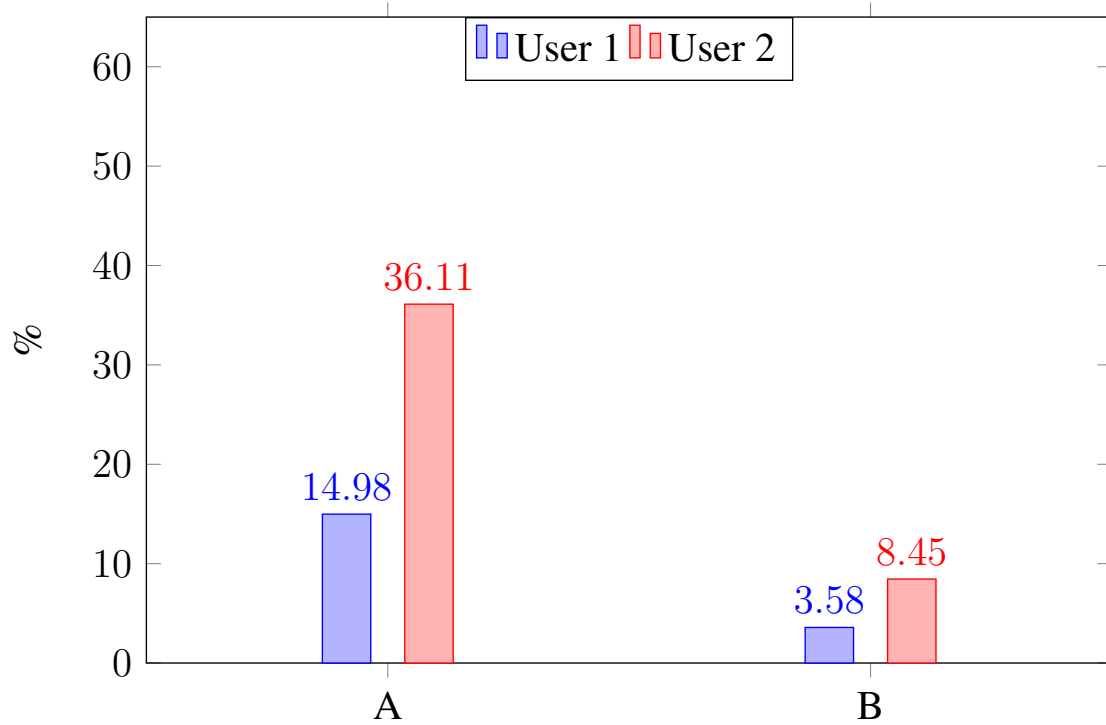


Figure 4.4: 15-word Word Error Rates. A: Non-Centered Hands, B: Centered Hands, No Wrist

CHAPTER 5

DISCUSSION

These results are highly promising. Given the small size of the training set (approximately 440 recordings of 36 words) and the relatively small size of the hands (approximately 256 pixels tall and wide), the high accuracy rate indicates that the current models could accommodate larger word sets with only a minor decrease in accuracy. While the models are inherently limited by their sole recognition of fingerspelling, the use of triletter HMMs opens the recognition of words outside the trained word list given a sufficiently-diverse recorded word list, which could expand the use of this model. However, it must be noted that these models can only recognize fingerspelling. While fingerspelling is a core component of ASL, it cannot be used as a direct form of communication. Nevertheless, in areas such as text entry, fingerspelling for many deaf users may be simpler to use than typing, and in this context, the models shown could be used in text entry systems, in much the same way that voice recognition systems are used. Furthermore, given that the models performed well against challenging data (rapidly moving, relatively small hands) when compared to data that could be collected on mobile devices (relatively-stationary, larger hands that occupy more of the recorded frame), the current model techniques are likely to translate well on a mobile device.

In regards to PopSign, an education game currently being developed for hearing adults to learn sign language on mobile devices, at each stage of the game, the user is presented with a small number (currently five) signs or words to recognize. Previously, due to the lower accuracy of recognition with non-centered finger joint position features, we considered choosing words to be grouped into stages based on each group's individual recognition accuracy. Based on the tentative results of our current model's given a large word list, the selection of words to be used at each stage of the game can be chosen primarily based on topic or relevancy, rather than purely sign language accuracy. This opens the door to more complex gameplay, such as more than five signs per stage.

CHAPTER 6

CONCLUSION

In this paper, we have shown that fingerspelling recognition is not only possible but highly accurate given mobile-like conditions based on resolution and hand size using limited data sets and trilleter HMMs. In user-dependent contexts, using centered hand and finger joint position data in addition to absolute data reduced the effects of movement of the hand between and within recordings on recognition by enhancing the model's ability to identify specific hand shapes. The models produced were sufficiently accurate on a large corpus of 36 words to be usable within educational games and learning tools.

While this work has shown the viability of the described models in accurate fingerspelling recognition in user-dependent contexts, further testing will be required to assess its accuracy when trained for user-independent contexts. Furthermore, given the extremely high accuracy of models, the systems designed can serve as a foundation for further development and usage in more complex settings, such as on mobile phones and low-resolution recordings. Finally, although our models have been shown to be highly accurate on offline data, the accuracy of such models on live data remains to be seen. Incorporating these models into games and testing their accuracy in this context would provide insight into the robustness of these models.

APPENDIX A
WORD LISTS

arm big box cow cup get hit off old pen pig
red see sky zoo blue book break doll fast give home
juice jump love oven pizza pony radio rock slide stove swing
trash wash zipper

Figure A.1: 36-word Dataset

REFERENCES

- [1] T. Supalla, “The classifier system in american sign language,” *Noun classes and categorization*, vol. 7, pp. 181–214, 1986.
- [2] H. Brashear, V. Henderson, K.-H. Park, H. Hamilton, S. Lee, and T. Starner, “American sign language recognition in game development for deaf children,” in *Proceedings of the 8th International ACM SIGACCESS Conference on Computers and Accessibility*, 2006, pp. 79–86.
- [3] H. R. V. Joze and O. Koller, “Ms-asl: A large-scale data set and benchmark for understanding american sign language,” *arXiv preprint arXiv:1812.01053*, 2018.
- [4] J. Huang, W. Zhou, Q. Zhang, H. Li, and W. Li, “Video-based sign language recognition without temporal segmentation,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, 2018.
- [5] N. C. Camgoz, O. Koller, S. Hadfield, and R. Bowden, “Sign language transformers: Joint end-to-end sign language recognition and translation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2020.
- [6] *Signall lab prices*.
- [7] R. I. Mayberry and E. B. Eichen, “The long-lasting advantage of learning sign language in childhood: Another look at the critical period for language acquisition,” *Journal of memory and language*, vol. 30, no. 4, pp. 486–512, 1991.
- [8] P. M. Brown and A. Cornes, “Mental health of deaf and hard-of-hearing adolescents: What the students say,” *Journal of deaf studies and deaf education*, vol. 20, no. 1, pp. 75–81, 2015.
- [9] R. E. Perkins-Dock Ph D, T. R. Battle MS, J. M. Edgerton MS, and J. N. McNeill MS, “A survey of barriers to employment for individuals who are deaf,” *JADARA*, vol. 49, no. 2, p. 3, 2015.
- [10] P. M. Sullivan and J. F. Knutson, “Maltreatment and disabilities: A population-based epidemiological study,” *Child abuse & neglect*, vol. 24, no. 10, pp. 1257–1273, 2000.
- [11] O. Turner, K. Windfuhr, and N. Kapur, “Suicide in deaf populations: A literature review,” *Annals of General Psychiatry*, vol. 6, no. 1, p. 26, 2007.
- [12] D. Warchoł, T. Kapuściński, and M. Wysocki, “Recognition of fingerspelling sequences in polish sign language using point clouds obtained from depth images,” *Sensors*, vol. 19, no. 5, p. 1078, 2019.

- [13] Y. Hu, H.-F. Zhao, and Z.-G. Wang, "Sign language fingerspelling recognition using depth information and deep belief networks," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 32, no. 06, p. 1 850 018, 2018.
- [14] C. K. Mummadi, F. P. P. Leo, K. D. Verma, S. Kasireddy, P. M. Scholl, and K. Van Laerhoven, "Real-time embedded recognition of sign language alphabet fingerspelling in an imu-based glove," in *Proceedings of the 4th international Workshop on Sensor-based Activity Recognition and Interaction*, 2017, pp. 1–6.
- [15] B. Dorner and E. Hagen, "Towards an american sign language interface," in *Integration of Natural Language and Vision Processing*, Springer, 1994, pp. 143–161.
- [16] C. Dong, M. C. Leu, and Z. Yin, "American sign language alphabet recognition using microsoft kinect," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, https://www.cv-foundation.org/openaccess/content_cvpr_workshops_2015/W15/html/Dong_American_Sign_Language_2015_CVPR_paper.html, 2015, pp. 44–52.
- [17] J. R. Pansare and M. Ingle, "Vision-based approach for american sign language recognition using edge orientation histogram," in *2016 International Conference on Image, Vision and Computing (ICIVC)*, <https://ieeexplore.ieee.org/document/7571278>, IEEE, 2016, pp. 86–90.
- [18] L. Ge, H. Liang, J. Yuan, and D. Thalmann, "3d convolutional neural networks for efficient and robust hand pose estimation from single depth images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, https://openaccess.thecvf.com/content_cvpr_2017/html/Ge_3D_Convolutional_Neural_CVPR_2017_paper.html, 2017, pp. 1991–2000.
- [19] O. Koller, H. Ney, and R. Bowden, "Deep hand: How to train a cnn on 1 million hand images when your data is continuous and weakly labelled," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, https://openaccess.thecvf.com/content_cvpr_2016/html/Koller_Deep_Hand_How_CVPR_2016_paper.html, 2016, pp. 3793–3802.
- [20] N. C. Camgoz, O. Koller, S. Hadfield, and R. Bowden, "Sign language transformers: Joint end-to-end sign language recognition and translation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, https://openaccess.thecvf.com/content_CVPR_2020/html/Camgoz_Sign_Language_Transformers_Joint_End-to-End_Sign_Language_Recognition_and_Translation_CVPR_2020_paper.html, 2020, pp. 10 023–10 033.
- [21] J. Hou *et al.*, "Signspeaker: A real-time, high-precision smartwatch-based sign language translator," in *The 25th Annual International Conference on Mobile Computing and Networking*, <https://dl.acm.org/doi/abs/10.1145/3300061.3300117>, 2019, pp. 1–15.

- [22] K. A. Weaver *et al.*, “Improving the language ability of deaf signing children through an interactive american sign language-based video game,” 2010.
- [23] Z. Zafrulla, H. Brashear, P. Yin, P. Presti, T. Starner, and H. Hamilton, “American sign language phrase verification in an educational game for deaf children,” in *2010 20th International Conference on Pattern Recognition*, IEEE, 2010, pp. 3846–3849.
- [24] P. Yin, I. Essa, T. Starner, and J. M. Rehg, “Discriminative feature selection for hidden markov models using segmental boosting,” in *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2008, pp. 2001–2004.
- [25] D. Bansal *et al.*, “Copycat: Using sign language recognition to help deaf children acquire language skills,” in *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, ser. CHI EA '21, Yokohama, Japan: Association for Computing Machinery, 2021, ISBN: 9781450380959.