ASLScribe: Real-Time American Sign Language Alphabet Image Classification Using MediaPipe Hands and Artificial Neural Networks

Alexander Costa **ALEXANDER.COSTA@UGA.EDU**
Department of Computer Science, University of Georgia

ABSTRACT

Sign language is an important form of communication for people with impaired hearing and/or speaking abilities. One type of sign language, American Sign Language (ASL), is used by approximately 500,000 people in the United States and is also used in Canada, Mexico, and many other countries. Among thousands of other signs, ASL consists of unique poses or gestures for each letter of the English alphabet. Image classification of these 26 signs constitutes a significant and challenging task due to the complexity of ASL alphabet signs, high interclass similarities, large intraclass variations, and frequent self-occlusion of the hand. This work presents a method for ASL alphabet classification using MediaPipe Hands, a high-fidelity hand and finger tracking model, and a machine learning classifier, as well as the ASLScribe program for transcribing ASL alphabet signs to text in real-time. Specifically, we investigate the performance four different machine learning classifiers (logistic regression, artificial neural networks, support vector machines, and k-nearest neighbors) on the 3D coordinates of 21 different hand and finger landmarks provided by MediaPipe Hands at the task of ASL alphabet sign recognition and use one of the strongest classifiers for transcribing ASL alphabet signs in ASLScribe. To train our models, the ASL Alphabet dataset by Akash Nagaraj, consisting of 87,000 ASL alphabet sign images, was used. The ASL Alphabet Test dataset by Dan Rasband, consisting of 870 alphabet sign images, was used for model evaluation, as well. The results indicate that a support vector machine classifier, which achieved the overall highest testing accuracy of 0.9095, is best suited to the task of classifying ASL alphabet sign images from hand and finger landmarks, though all the models investigated were found to achieve an acceptable testing accuracy greater than 0.85. Support vector machines and logistic regression consistently performed better than artificial neural networks and k-nearest neighbors.

Index Terms: *American sign language, sign language recognition, sign language transcription, image classification, artificial neural networks, logistic regression, support vector machines, k-nearest neighbors, ASL, ANN, SVM*

I. INTRODUCTION

BACKGROUND INFORMATION

American Sign Language is a critically important mode of communication for individuals with impaired hearing or speaking abilities as it allows for communication through visual cues alone. It is widely used throughout the United States as well as Canada, Mexico, and many other countries [1]. Despite its prevalence, communication between ASL users and non-sign-language speakers is still a considerably difficult problem. While professional interpreters exist, they are often not readily available and costly. An automatic ASL recognition system could constitute a remarkable and especially valuable advancement in the image recognition field and the sign language world at large. Not only could such a system ameliorate the difficulties of communication between ASL and non-ASL speakers but could facilitate numerous advancements in the intersection between ASL and human-computer interaction.

For decades, researchers have tried to solve the challenging problem of sign language recognition. Many proposed solutions rely on external devices, such as depth cameras, sensors, gloves, or motion capturing systems [1, 2]. Such constraints limit the applicability of these solutions to environments where these tools are available. Recently, given the demonstrated success of deep learning methods on computer vision tasks, focus has shifted to purely vision-based sign recognition powered by deep learning techniques. Being non-intrusive and requiring only a basic phone camera or webcam to generate input, these methods have the potential to be significantly more applicable and wide-reaching than solution requiring external devices. Deep learning-based solutions, however, suffer from a lack of large-scale, public sign language databases suitable for machine learning, as well as a weakened ability to differentiate between interclass similarities and increased susceptibility to self-occlusions of the fingers or hands.

In this work, we address the need for a robust sign language recognition system by focusing on the simpler, though still challenging, task of ASL alphabet image recognition and real-time transcription using MediaPipe Hands and an artificial neural network (ANN) classifier.

RELATED WORKS

Since the 1980s, researchers have tried to solve the problem of ASL recognition in many different ways. One of the first approaches, which has been reapproached may times over the years, relies on the use of a sensor glove that can track hand and finger movements [4, 5, 6, 7, 8]. Another approach was vision-based recognition, beginning in 1988 [9]. Since signing takes place in three dimensions, many researchers following a vision-based approach used depth information [10, 11] or multiple cameras [12].

In this paper, we address the need for a nonintrusive sign language recognition technique which could prove to be far more accessible and generalizable than methods which rely on external hardware. The use of only RGB channels for sign recognition has been approached before with traditional computer vision techniques [13, 14]. More recently, deep learning techniques have also been applied to the sign recognition problem with 2D- [15] and 3D-CNNs [3, 16].

Most approaches struggle with a tradeoff between high classification accuracies and breadth of signs recognized.

RESEARCH OBJECTIVES

Aim 1 *– Construct a machine learning model which can accurately classify ASL alphabet signs in real-world environments.*

Aim 2 *– Construct a program capable of transcribing ASL alphabet signs into text in real-time.*

Our primary research objectives are summarized in *Aim 1* and *Aim 2* above. To achieve these objectives, we proceeded in a sequential, two-phased approach. In phase 1, we investigated the performance of four different machine learning classifiers – logistic regression (LR), artificial neural networks (ANN), support vector machines (SVM), and k-nearest neighbors (KNN) – at the task of ASL alphabet sign classification using 21 3D hand and finger landmark coordinates provided by MediaPipe Hands. In phase 2, we constructed a program using one of the strongest of the investigated classifiers which transcribes ASL alphabet signs into text in real-time. During phase 1, the four models were each constructed and trained on our training dataset. Each model performs a multiclass classification of ASL alphabet sign testing images as one of 28 classes, including the 26 English alphabet letters and 2 auxiliary classes, SPACE, DELETE (which are helpful for the real-time transcription program). After training of the models, each model is subject to a comparison of top-1 accuracies on an unseen training dataset as an evaluation of model performance.

The datasets used to train and test our models are the ASL Alphabet dataset by Akash Nagaraj and the ASL Alphabet Test dataset by Dan Rasband, respectively. These datasets consist of labeled 200×200-pixel 3-channel RGB ASL alphabet sign images which comprise 28 classes (26 for the letters

A-Z and the two aforementioned auxiliary classes). Our unprocessed training data consists of 84,000 images with 3,000 images per class, and our unprocessed testing data consists of 870 images with 30 images per class. Both datasets are perfectly balanced, eliminating the need for any dataset balancing. Before training, each image is fed into the MediaPipe Hands model, yielding 21 3D hand and finger landmark coordinates which are then used for training and testing of our models. Due to the poor lighting conditions of some training and test images, MediaPipe Hands did not recognize the existence of a hand in some input images. Any training or test images for which MediaPipe Hands did not recognize a hand were thrown out. This process yielded 61,115 training instances and 807 testing instances.

After the preprocessing of our image sets, we perform four identical experiments for each of the four classifiers investigated in this research. First, the training and testing data of 21 3D hand and finger landmark coordinates was split into two copies, wherein the coordinates of the first are raw proportions of the width and height of the input image and the coordinates of the second are normalized proportions of the width and height of the hand bounding box. Second, each model is trained with the training dataset and then evaluated against the testing dataset yielding the top-1 accuracy scoring metric for each model.

The results indicate that a support vector machine classifier, which achieved the overall highest testing accuracy of 0.9095, is best suited to the task of classifying ASL alphabet sign images from hand and finger landmarks. The top two performing models were the LR (accuracy = 0.8860) and ANN (accuracy = 0.8797) for raw input data and the SVM (accuracy = 0.9095) and LR (accuracy = 0.8959) for normalized input data. All models except KNN performed well on raw input data. After normalization, all models except ANNs, whose testing accuracy actually fell by one percentage point, saw an increase in performance, especially KNNs, whose testing accuracy increased by over 17 percentage points.

The remainder of this paper is broken into a number of sections. Section II describes our training and test datasets and discloses our data preprocessing techniques in further detail. Section III describes our experiments in further detail and provides our experimental results in tabular form. Section IV presents an analysis of our experimental results and the confusion matrices of each of our best models, Section V presents a short discussion about ASLScribe and finally section VI provides a conclusion and discussion of the paper at large.

II. DATA

INTRODUCTION

The datasets used in this work were based on the American Manual Alphabet (AMA). Figure 1 shows all hand positions of the alphabet.

Figure 1. Hand poses used for constructing each letter in the American Manual Alphabet.

The datasets used for the training and testing of our models were comprised of static hand poses of each letter from the alphabet, as well as hand poses for SPACE and DELETE classes which are useful for the ASLScribe real-time transcription program. An

additional NOTHING class, consisting of only background images, was present in our training and testing dataset, but was not used for training or testing since the presence of a hand is already reported by the MediaPipe Hands model. Gesture-based signs, J and Z, were included in the training and testing datasets for completeness despite their temporal dimension. Static poses of these gesture-based signs taken at different times along the gesture sequence were used in both training and testing.

The training data, the ASL Alphabet dataset by Akash Nagaraj, consists of 84,000 200×200-pixel 3-channel RGB ASL alphabet sign images which comprise 28 classes (26 for the letters A-Z and the two aforementioned auxiliary classes). Each of the 28 classes contains 3,000 instances making a perfectly balanced dataset. Figure 3 displays example training images from the ASP Alphabet dataset.

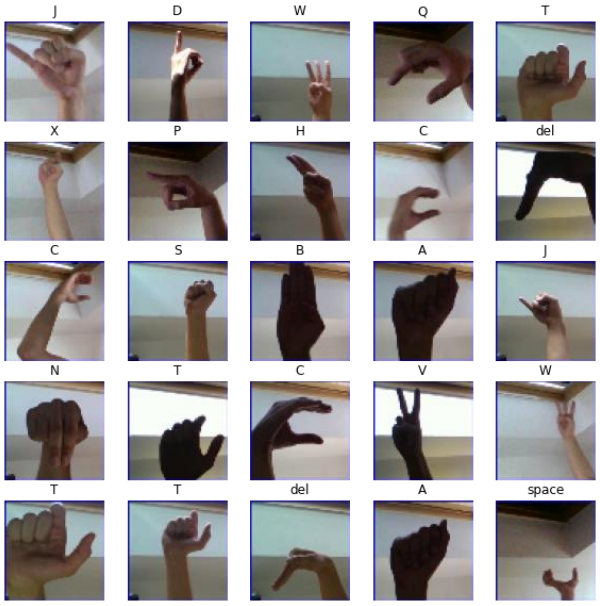

Figure 2. Example training images from the ASL Alphabet dataset by Akash Nagaraj with the ground truth class label located above each image. Of note is the varied lighting and positioning of each pose, but lack of signer and background variation.

Though the training data does provide variation in the form of lighting schemes and relative positioning, it still suffers greatly from excessive homogeneity. All images were taken by the same signer against a relatively static background. In ASL signing, there often exists subtle variation in positioning between different signers, not to mention regional and cultural differences which manifest as positioning variation between signers. With signer dependency being one of the most blocking challenges of current non-intrusive sign recognition approaches, the lack of multiple signer representation in the training data constitutes an enormous drawback to the generalizability of this training data. Furthermore, the lack of background variation in our training data constitutes yet another drawback to generalizability. To ameliorate some of these intrinsic shortcomings of the training dataset, we train our models not on the pixel color values of each input image but rather the 3D coordinates of 21 hand and finger landmarks provided by the MediaPipe Hands model. By doing so, our models are not concerned with any lighting, background, or color information and are instead informed purely by the position and orientation of the hand and fingers. The homogeneity of the dataset no longer poses such a strong detriment to our models' generalizability since our models may focus entirely on primary class discerning information instead of secondary or auxiliary information like colors, lighting, and backgrounds.

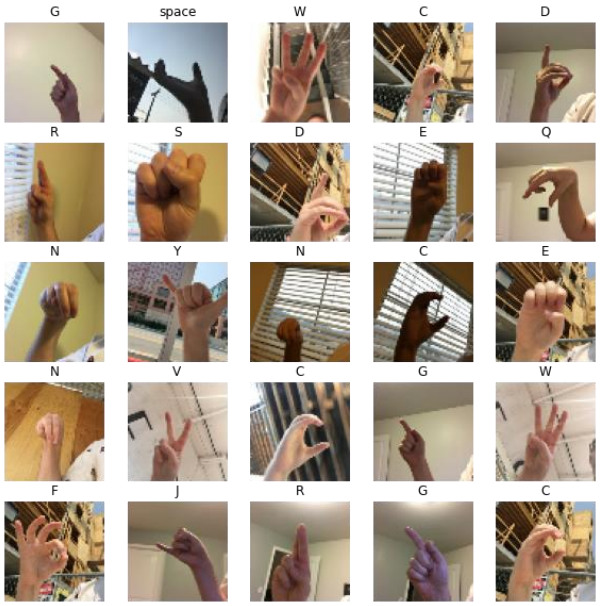

Figure 3. Example testing images from the ASL Alphabet Test dataset by Dan Rasband with the ground truth class label located above each image. Of note is the varied lighting and positioning of each pose, but lack of signer and background variation.

The testing data, the ASL Alphabet Test dataset by Dan Rasband, consists of 840 200×200-pixel 3-channel RGB ASL alphabet sign images which comprise the same 28 classes as the training data. Each of the 28 classes contains 30 instances making a perfectly balanced dataset. The testing data also makes the effort to use varied lighting schemes, relative positioning, and backgrounds but still suffers from the same homogeneity issues which affect the training data. Figure 3 displays example testing images from the ASL Alphabet Test dataset.

Both testing and training data were shared on Kaggle.com for the express purpose of ASL alphabet image classification.

PREPROCESSING

The training and testing datasets were quite clean and required little preprocessing. Training and testing images were fed directly into the MediaPipe Hands model for mapping to 3D coordinates of 21 hand and finger landmarks. Due to the poor lighting conditions of some training and test images, MediaPipe Hands did not recognize the existence of a hand in some input images. Any training or test images for which MediaPipe Hands did not recognize a hand were thrown out. This process yielded 61,115 training instances and 807 testing instances. Class frequencies were still very similar, so no dataset balancing was pursued. Since each of the 21 hand and finger landmarks was composed of an X, Y, and Z coordinate, each test or training instance was flattened into a 63-feature input space composed of the X, Y, and Z coordinate of all 21 hand and finger landmarks. Figure 4 provides a diagram of each of the 21 landmarks. Figures 5 and 6 display the 21 hand and finger landmarks overlaying their respective hand for the training and testing datasets.

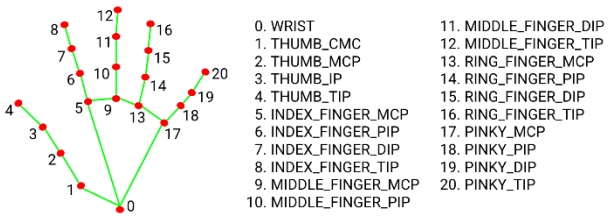

Figure 4. The 21 hand and finger landmarks.

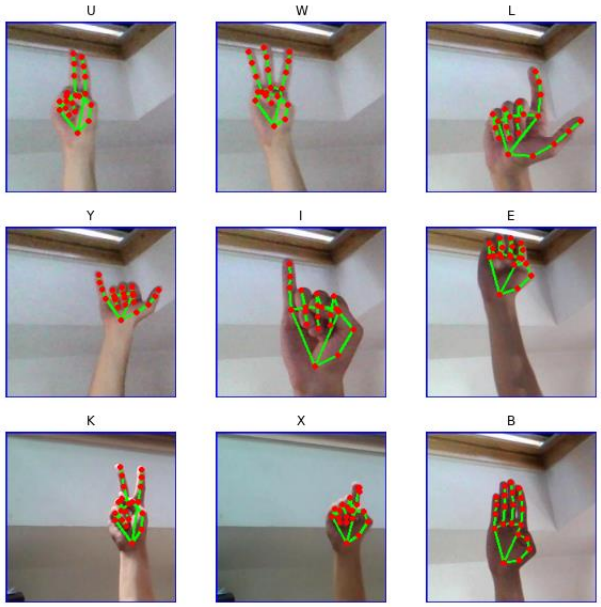

Figure 5. Hand and finger landmarks of the training set superimposed over their respective hands. Ground truth class labels are given above each image.

Figure 6. Hand and finger landmarks of the testing set superimposed over their respective hands. Ground truth class labels are given above each image.

PARTITIONING

Raw landmark X and Y coordinates are given as a proportion of the width and height of the input image, respectively. Raw landmark Z coordinates are estimated relative depths. Since raw X and Y coordinates are dependent upon the position of the hand within the input image, the data is not invariant

to translation or scaling. An "A" sign that appears in the bottom left corner of the input image and an "A" sign that appears in the top right corner of the input image will have drastically different X and Y coordinates. With such high intraclass variability, our models are likely to suffer. To ameliorate this issue, we split the training and test datasets into two copies. The first of which contains raw coordinates based on the width and height of the input image, and the second of which contains normalized coordinates based on the bounding box surrounding the hand. More specifically, normalized X and Y coordinates are given as a proportion of the width and height of the bounding box surrounding the hand and fingers, respectively. The bounding box is calculated using the minimum and maximum landmark in the raw X and raw Y dimensions. The Z coordinate is not altered. These normalized coordinates are invariant to translation or scaling, so the coordinates of any sign of the same letter should be relatively similar. With intraclass variability minimized, our models may be able to perform better.

Evaluation of models was performed by testing models with the aforementioned ASL Alphabet Test dataset consisting of signs performed by a different signer in foreign environments. The motivation behind evaluating models with such disparate data is to emphasize the goal of generalizability. It is a rather trivial and ineffectual problem to classify signs performed by a single signer in a static environment. To classify signs performed by many signers in varied environments, however, is a much more valuable endeavor, so the dissimilar testing data is used to encourage our models to search for generalizable solutions.

III. EXPERIMENTS

For our experiments, we used the Python programming language, Scikit-Learn and TensorFlow machine learning toolkits and Keras neural network library for data preprocessing, model training and evaluation. Experiments were performed on a cloud computing engine with 16GB RAM and 2-core Intel Xeon CPU.

Following data preprocessing we constructed four different classifiers: a logistic regression classifier, an artificial neural network classifier, a support vector machine classifier, and a k-nearest neighbors classifier. Table 1 presents an overview of each of these four models and their parameters.

Table 1: Investigated Classifiers

Name	Parameters	Toolkit
Logistic Regression	SAG solver, 1,000 max iterations	Scikit-Learn
Artificial Neural Network 3 fully connected hidden layers. *Input(63) → Fully Connected(128) → Fully Connected(96) → Fully Connected(64) → Output(28)*	ReLU activation function, ADAM optimizer, Categorical Cross-Entropy loss, 10 epoch training, 28,604 total parameters	TensorFlow
Support Vector Machine	RBF kernel	Scikit-Learn
K-Nearest Neighbors	K=15	Scikit-Learn

Table 1. The four investigated classifiers with accompanying parameters. A number within parentheses following a layer name indicate the output dimension in the case of Fully Connected Layers. All models were instantiated with the default parameters of their respective toolkit unless otherwise noted in the Parameters column.

Following training, each model was tested against hand and finger landmarks of the ASL Alphabet Test dataset for an estimation of model performance. The results of this final test for each model are given in Table 2. No model took a prohibitive amount of time to complete training or testing.

Table 2: Experimental Results for the Four Classifiers					
Input Data	**Name**	**Training Accuracy**	**Testing Accuracy**	**Training Time (s)**	**Testing Time (s)**
Raw	**Logistic Regression**	**0.9216**	**0.8860**	**30.05**	**0**
	Artificial Neural Network	0.9568	0.8797	22.92	0
	Support Vector Machine	0.9439	0.8761	120.16	0
	K-Nearest Neighbors	0.9407	0.7113	2.86	0
Normalized	Logistic Regression	0.9316	0.8959	51.14	0
	Artificial Neural Network	0.9586	0.8634	23.20	0
	Support Vector Machine	**0.9602**	**0.9095**	**55.25**	**0**
	K-Nearest Neighbors	0.9629	0.8835	2.56	0

Table 2. Experimental results for all four classifiers. The models achieving the highest test accuracies for each input data type are bolded.

The results indicate that a support vector machine classifier, which achieved the overall highest testing accuracy of 0.9095, is best suited to the task of classifying ASL alphabet sign images from hand and finger landmarks. The top two performing models were the LR (accuracy = 0.8860) and ANN (accuracy = 0.8797) for raw input data and the SVM (accuracy = 0.9095) and LR (accuracy = 0.8959) for normalized input data.

Practically speaking, logistic regression, artificial neural networks, and support vector machines all performed well on raw input data though SVMs consistently took 4-5 times longer training time than the LR or ANN models. After normalization, all models except ANNs, whose testing accuracy actually fell by one percentage point, saw an increase in performance, especially KNNs, whose testing accuracy increased by over 17 percentage points. All models held consistently higher training accuracies than testing accuracies, as is expected. Accuracies for ANNs are averaged over 10 runs to account for accuracy perturbations due to chance.

IV. ANALYSIS

Unlike in [3], where testing accuracies greater than 0.50 were unachievable, we have easily and consistently achieved testing accuracies greater than 0.85 and one model even achieves an accuracy over 0.90. While these accuracies are acceptable for production use in ASLScribe, we believe higher testing accuracies may be achieved.

First and foremost, our training and testing datasets are markedly dissimilar from one another yet excessively homogenous within. As discussed in Section II, our training set consists of images from only one signer in a relatively static environment. This homogeneity does not seem to offer our models enough intraclass variability information in order for them to generalize as well as they could to other signers, which is demonstrated by our lower testing accuracies. Ideally, a training set consisting of signs by tens, if not hundreds, of different signers would be used to account for all kinds of real-world variability which our training set lacks. We hypothesize that a larger, more varied training dataset could offer improvements to our models' accuracies.

On top of an excessively homogenous training dataset, the classification of ASL alphabet signs itself presents many obstacles due to high interclass similarities. Many letter signs are extremely similar, differing only by one finger's position (G/H, K/V, U/V, E/S, M/N/S/T) or even simply the rotation of the hand (P/K, I/J, D/Z). Even a beginner ASL user might mistake one of these letters for the other. To see whether the investigated models were themselves susceptible to these interclass similarities, we investigate the confusion matrices of the best iteration of each model presented in Figure 7.

Counts falling along the main diagonal are correctly classified test images. Colored squares outside the main diagonal constitute significant and habitual mistakes in the model classification. Many significant misclassifications do in fact correspond with those class pairs having high interclass similarities. For example, all four models have a high confusion rate between Ms and Ns, which

differ only by thumb position. All four models struggle, as well, at discerning between Vs and Ks which also differ only in thumb position. Likewise, all four models habitually predict Cs as Os and DELETEs as Qs, which share many similarities.

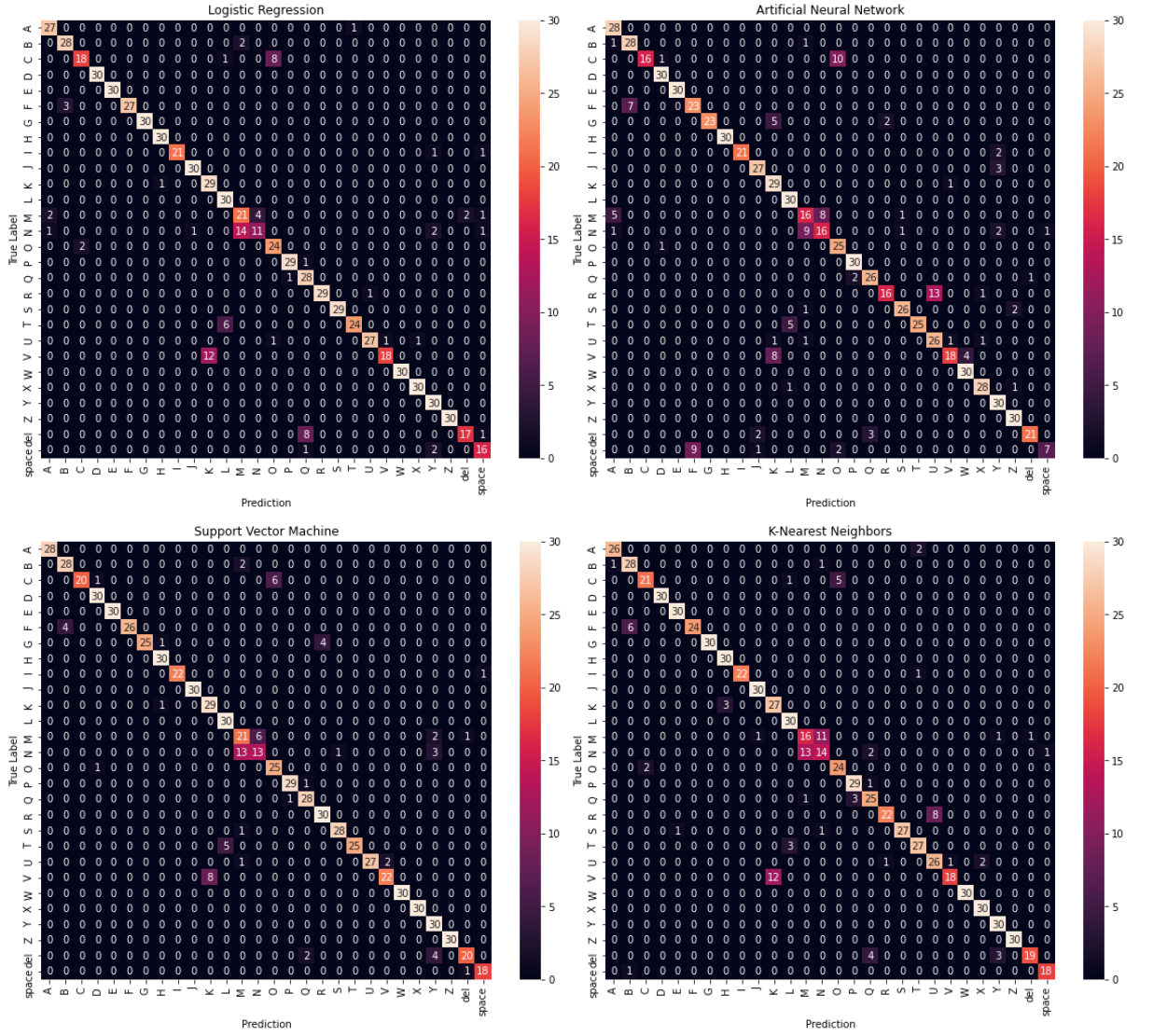

Figure 7. From left to right, top to bottom: The confusion matrices for the best logistic regression model, best artificial neural network model, best support vector machine model, and best k-nearest neighbors model.

From these abundant examples, it is clear that the test accuracies of the best models suffer due to the high interclass similarities between many classes. To accurately discern between such similar classes our models may need more hand and finger landmarks or a more accurate landmark tagging model to differentiate similar classes more easily through the increase in available and accurate information. Some domain specific heuristics could also be used to differentiate between these classes known to be very similar. For example, when differentiating between two classes of known similarity, we might add another pass to an algorithm to check a specific subset of landmarks for minute differences or apply a specific weighting scheme.

V. ASLSCRIBE

Our ASLScribe real-time ASL alphabet transcription program is a very simple extension of the work completed in this paper. Using the MediaPipe Hands JavaScript API and the

TensorFlow JavaScript API we supply a real-time feed of webcam frames to the MediaPipe Hands model which yields the 21 landmark coordinates discussed in Section II. These landmarks are then fed into the artificial neural network investigated in Section III which, in turn, yields a character prediction. Upon sign chance, the previous character is written to a buffer. Users also have the option to sign a DELETE or SPACE character which performs a delete operation or inserts a space character, respectively.

The artificial neural network classifier was chosen over the logistic regression and support vector machine classifiers even though the ANN has a worse testing accuracy performance because of the ease of use of the Tensorflow JavaScript API. The LR and SVM classifiers were implemented in Scikit-Learn, which lacks a JavaScript API.

The ASLScribe program is hosted on the authors personal website (https://alexcostaluiz.com/asl-classification/).

VI. CONCLUSION

Our work indicates that among many different simple machine learning classifiers, the support vector machine performs best at ASL alphabet sign classification using hand and finger landmark coordinates. All of the investigated models were able to achieve acceptable testing accuracies greater than 0.85, with the support vector machine classifier being the only model to achieve a testing accuracy greater than 0.90. Various hyperparameter tuning procedures for all models could not increase the testing accuracy beyond 0.90. It seems our models are mostly bottlenecked by the lack of signer variation in our training dataset and by high interclass similarities in the ASL alphabet.

Our findings present a number of opportunities for the experiments herein to be continued and extended. Firstly, the aggregation and preprocessing of many more ASL alphabet datasets in addition to those used in this work could be undertaken to provide the models with much more varied, real-world information and subsequently strengthen their generalizability.

Though a fair amount of hyperparameter tuning was attempted on each of the four models investigated, we feel it may prove useful to investigate more even model variations and even other classifiers (e.g., random forest classifier), in general.

Finally, a natural extension of the work presented in this paper would be to incorporate more ASL signs beyond the manual alphabet into the real-time ASLScribe program so that users can express themselves more freely and efficiently. Furthermore, developing a temporally sensitive model (e.g., recurrent neural networks) so that more complex gestures may be detected would be of great use.

REFERENCES

[1] 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.

[2] W. Tao, M. C. Leu, and Z. Yin, "American Sign Language alphabet recognition using Convolutional Neural Networks with multiview augmentation and inference fusion," *Engineering Applications of Artificial Intelligence*, vol. 76, pp. 202-213, 2018.

[3] A. L. Costa, "American Sign Language (ASL) Alphabet Image Classification Using Convolutional Neural Networks," (in English), p. 11, 04/12/2021 2021.

[4] Gary J. Grimes. Digital data entry glove interface device, November 1983. US Patent.

[5] C. Charayaphan and A. E. Marble. Image processing system for interpreting motion in American Sign Language. Journal of Biomedical Engineering, 14(5):419–425, September 1992. ISSN 0141-5425. doi: 10.1016/0141-5425(92)90088-3.

[6] S. S. Fels and G. E. Hinton. Glove-Talk: A neural network interface between a data-glove and a speech synthesizer. IEEE Transactions on Neural Networks, 4(1):2–8, January 1993. ISSN 1045-9227. doi: 10.1109/72.182690.

[7] Rung-Huei Liang and Ming Ouhyoung. A real-time continuous gesture recognition

system for sign language. In Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on, pages 558–567. IEEE, 1998.

[8] Syed Atif Mehdi and Yasir Niaz Khan. Sign language recognition using sensor gloves. In Neural Information Processing, 2002. ICONIP'02. Proceedings of the 9th International Conference on, volume 5, pages 2204–2206. IEEE, 2002.

[9] Shinichi Tamura and Shingo Kawasaki. Recognition of sign language motion images. Pattern Recognition, 21(4):343–353, 1988. ISSN 0031-3203. doi: 10.1016/0031-3203(88)90048-9.

[10] Alina Kuznetsova, Laura Leal-Taixé, and Bodo Rosenhahn. Real-time sign language recognition using a consumer depth camera. In Proceedings of the IEEE International Conference on Computer Vision Workshops, pages 83–90, 2013.

[11] Zahoor Zafrulla, Helene Brashear, Thad Starner, Harley Hamilton, and Peter Presti. American sign language recognition with the kinect. In Proceedings of the 13th international conference on multimodal interfaces, pages 279–286. ACM, 2011.

[12] Helene Brashear, Thad Starner, Paul Lukowicz, and Holger Junker. Using multiple sensors for mobile sign language recognition. Georgia Institute of Technology, 2003.

[13] Thad Starner and Alex Pentland. Real-time american sign language recognition from video using hidden markov models. In Motion-Based Recognition, pages 227–243. Springer, 1997.

[14] Thad E Starner. Visual recognition of american sign language using hidden markov models. Technical report, Massachusetts Inst Of Tech Cambridge Dept Of Brain And Cognitive Sciences, 1995.

[15] Oscar Koller, Sepehr Zargaran, Hermann Ney, and Richard Bowden. Deep Sign: Enabling Robust Statistical Continuous Sign Language Recognition via Hybrid CNN-HMMs. International Journal of Computer Vision (IJCV), 126(12):1311–1325, December 2018. ISSN 1573-1405. doi: 10.1007/s11263-018-1121-3.

[16] Jie Huang, Wengang Zhou, Houqiang Li, and Weiping Li. Sign language recognition using 3d convolutional neural networks. In Multimedia and Expo (ICME), 2015 IEEE International Conference on, pages 1–6. IEEE, 2015.