An AI-Based Approach To American Sign Language Alphabet Recognition

Author: Vaibhav Akella

ABSTRACT

This research presents the development of an AI-powered image recognition model designed to translate American Sign Language (ASL) alphabet gestures into corresponding text characters (A-Z). The models utilize machine learning algorithms, specifically random forest and multi-layer perceptron (MLP) Classifiers with logistic regression as a baseline, trained on a dataset of labelled hand gesture images to classify static signs representing individual letters accurately. Both models achieve high accuracy in identifying hand gestures under controlled lighting and background conditions in a short time frame, with the MLP achieving an accuracy of 98.79% and the random forest achieving a slightly higher accuracy of 99.38% while the baseline achieved an accuracy of 97.66%. These models are particularly beneficial for individuals who are hearing impaired or for those who wish to improve communication with ASL users. Future improvements may include expanding to dynamic gestures as well as recognizing full words or phrases.

1. INTRODUCTION

Communication is a vital component of human interaction, yet individuals who are deaf or hard of hearing often encounter obstacles in everyday settings where sign language is not widely understood. American Sign Language (ASL) offers a powerful means of expression, but its effectiveness depends on the presence of a mutual understanding between parties. Given that many hearing individuals are not fluent in ASL, this presents a significant barrier to communication. This underscores the need for solutions to surmount this challenge.

The increasing capabilities of artificial intelligence (AI) present new opportunities to bridge this communication gap. In recent years, machine learning has transformed the field of image recognition. While traditional methods often require domain-specific knowledge to design features, machine learning models learn patterns directly from raw data, bypassing the need for

manual feature engineering (Temitope, Nguyen Thanh, & Victor, 2025). In this paper, we leverage these methods to design a system that automatically interprets ASL gestures and translates them into readable text, providing real-time assistance for those who rely on sign language. Specifically, we employ random forests, a type of ensemble classifier, and multi-layer perceptrons (MLPs), a type of neural network. The random forest algorithm, introduced by Breiman (2001), is a robust ensemble method that builds multiple decision trees, described by Rokach & Maimon as “a classifier expressed as a recursive partition of the instance space” (2005), and combines their predictions. It is known for its high performance and resistance to overfitting, particularly in high-dimensional spaces. On the other hand, the MLP classifier is a type of feedforward neural network that works using multiple hidden layers and activation functions (Bishop, 2006). MLPs are especially effective in scenarios where data exhibits intricate spatial or visual patterns, as is the case with ASL hand gestures (Mohammadi et al., 2022).

Sign language recognition, specifically for the ASL alphabet, has been a topic of active research. While some approaches utilize deep convolutional neural networks (CNNs) (Pigou et al., 2015), this study focuses on more parsimonious models that require fewer parameters and less computational power, making them better suited for deployment on resource-constrained or embedded devices When paired with careful preprocessing and hyperparameter tuning, the machine learning models tested in this study - logistic regression, random forests, and MLPs – still achieve competitive performances, as measured by mean accuracy scores of 0.9766, 0.9983 and 0.9879 respectively.

In this paper, we implement a solution that enables real-time ASL translation in a highly accurate manner. This approach prioritizes model efficiency and ease of implementation, making it more broadly accessible than alternative methods.

2. METHODOLOGY

2.1 DATASET INFORMATION

The dataset employed for this project is the "ASL Alphabet" dataset found on Kaggle.com (Akash Nagaraj, 2018), which consists of labelled grayscale images representing the 26 letters

of the ASL alphabet as well as ‘del’ and ‘space.’ In addition, negative controls of images containing no letters are also included. Specifically, each image depicts a static hand gesture corresponding to a single letter and is stored in folders based on the letter category. The dataset is well-structured, with each class containing thousands of images captured under consistent lighting and backgrounds.

For the purpose of this study, subsets containing 200 images per category were used to assess the scalability and learning behaviour of the models. All images were resized to 64×64 pixels and flattened into one-dimensional arrays for compatibility with the classifiers. Notably, only the “Training” dataset on the website was used. These were subsequently split into new training and testing sets using a 3:1 ratio.

2.2 PREPROCESSING

Effective preprocessing is critical in image recognition tasks. In this study, each image was first converted to grayscale to reduce computational complexity without sacrificing gesture-specific features. The images were then resized to a uniform dimension of 64×64 pixels and flattened. To normalize the pixel intensities, each value was divided by 255.0, bringing all inputs into the [0, 1] range. This normalization improves model convergence during training.

The data was split into training and testing subsets with a fixed random seed to ensure reproducibility.

2.3 MODELS

Three machine learning classifiers were implemented:

- **Logistic classification**: This is our baseline model, intended to provide a point of comparison against the more complex MLP and Random Forest classifiers. We implemented multinomial Logistic Regression using the LBFGS solver, which is well-suited for handling multi-class classification problems, and a maximum number of iterations of 500 to ensure convergence given the size and dimensionality of the dataset.

- **Multi-layer Perceptron (MLP) Classifier**: This model used two hidden layers with 150 and 150 neurons respectively, a batch size of 128, a learning rate of 0.001, and a maximum of 500 training iterations. These parameters were chosen based on research compiled through experimentation (eg, Figure 2.3.1).
- **Random Forest Classifier**: Configured with 100 decision trees and a fixed random seed, this ensemble method was chosen for its interpretability, robustness to overfitting, and high performance on structured data (Breiman, 2001).

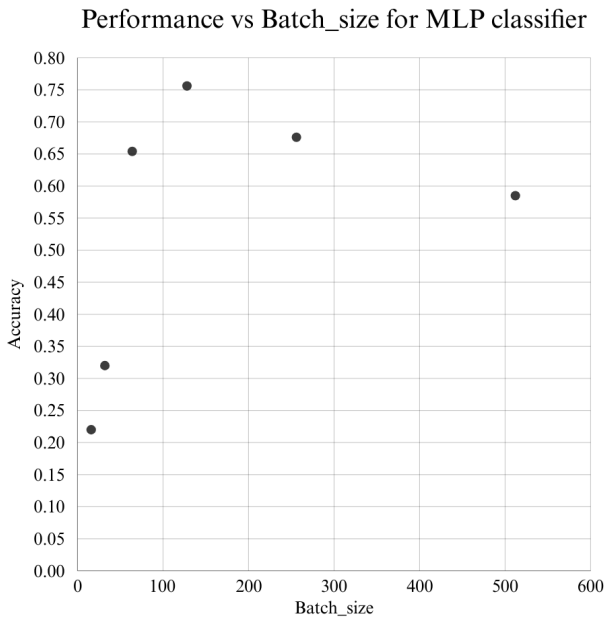

Figure 2.3.1: Accuracy vs. batch size curve for the MLP classifier, where a size of 128 had the best accuracy.

2.4 HYPERPARAMETER TUNING

To further optimize performance, several experiments were conducted by varying key hyperparameters:

- For Logistic Regression, the maximum number of iterations was tested in the range 100 - 1000.
    - There was a relatively upward trend as the maximum number of iterations increased; however, as the max_iter increased, the time taken to classify also

increased. Thus, due to the minimal increase in accuracy past 500 iterations, around 97%, 500 iterations was finally chosen.

- For Random Forest, the number of estimators was tested in the range of 50 to 400.
    - The best results came when n_estimators = 100, tested across multiple random states and varied numbers of images; 100 estimators consistently had an accuracy of around 99%. On the other hand, the other values tested would either lead to lower accuracy or far too much time taken.

- For MLP, variations included the number of hidden layers, batch sizes, learning rates, and maximum iteration counts.
    - Hidden layer sizes, the values of the parameters tested were:
        - 1 hidden layer with 50 neurons
        - 1 hidden layer with 100 neurons
        - 2 hidden layers with 100 neurons each
        - 2 hidden layers with 150 neurons each
        - 2 hidden layers with 200 neurons each
        - 3 hidden layers with 100, 100, and 100 neurons respectively
        - 3 hidden layers with 150, 150, and 150 neurons respectively
    - From these experiments, 2 hidden layers with 150 neurons each had the best overall accuracy, while simultaneously providing the least tradeoff in terms of time taken.

    - For batch sizes, the values tested were 64, 128, and 256, where 128 showed the greatest accuracy (Figure 2.3.1).

    - The learning rate showed an inverse relationship with accuracy, but if too low, it would lead to the classifier only predicting one letter; the best performance was when the learning rate was approximately 0.001.

- Maximum iterations showed a rather linear relationship with accuracy; however, if it is too high, it would lead to the time taken to increase drastically. 500 was found to be a good compromise between accuracy and time taken (Figure 2.4.1).

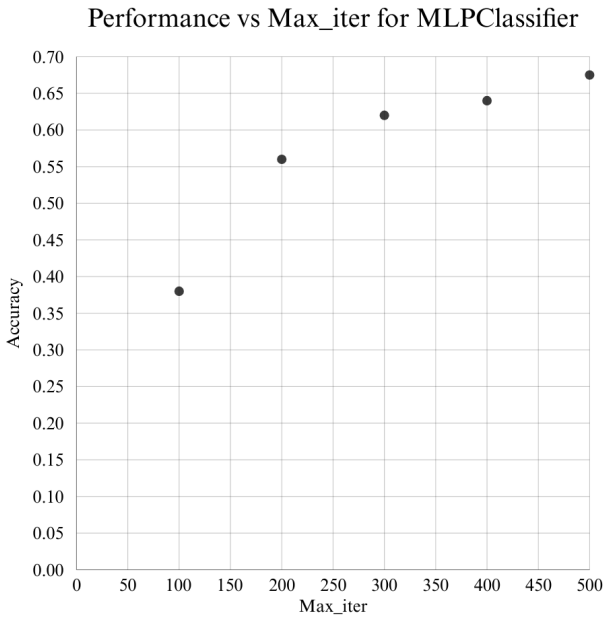

Figure 2.4.1 *Accuracy vs. Max_iter for the MLP classifier, where a max of 500 had the best accuracy.*

Each configuration was trained and evaluated, with accuracy metrics recorded and visualized to identify optimal settings.

2.5 EVALUATION METRICS

Model performance was evaluated using classification accuracy, the area under the receiver operating characteristic curve (AUROC), the area under the precision-recall curve (AUPRC), and confusion matrices. These matrices provided insight into specific misclassification patterns, such as common errors between visually similar letters in the ASL alphabet (e.g., W and V, or S and Z).

3. RESULTS

The performance evaluation revealed minor differences between models, both in mean accuracy and in stability as reflected by standard deviation measures. The logistic regression baseline achieved a mean accuracy of 0.9838 (SD = 0.0051) for 5-fold cross-validation and a mean accuracy of 0.9803 (SD = 0.0035) when bootstrapped. The MLPClassifier achieved a mean accuracy of 0.9869 (SD = 0.0090) under 5-fold CV and 0.9851 (SD = 0.0042) with bootstrapping, indicating great predictive performance on this task. Likewise, the RandomForestClassifier demonstrated near-perfect classification ability, with 5-fold CV results of 0.9955 (SD = 0.0018) for accuracy. Bootstrapping produced comparable results, with a mean accuracy of 0.9945 (SD = 0.0022). In single-train/test evaluations, the logistic regression baseline also performed strongly (Accuracy = 0.9766, AUROC = 0.9992, AUPRC = 0.9953), though it was slightly outperformed by both the MLP (Accuracy = 0.9879, AUROC = 0.9992, AUPRC = 0.9953) and random forest (Accuracy = 0.9983, AUROC = 1.0000, AUPRC = 0.9996). The exceptionally high AUROC and AUPRC scores for the Random Forest model indicate not only an ability to correctly separate classes but also a robustness to threshold selection, suggesting minimal overlap between class distributions in the feature space.

The classifiers demonstrated strong performance across all experimental configurations. The results are summarised below in Table 3.1.

Model	Accuracy	AUROC	AUPRC	5-fold CV	Bootstrapping
Logistic regression	0.9766	0.9992	0.9953	0.9838	0.9803
MLP Classifier	0.9879	0.9994	0.9973	0.9869	0.9851
Random forest	0.9983	1.0000	0.9996	0.9955	0.9945

Table 3.1 Table containing the corresponding accuracy, AUROC, and AUPRC scores for each model.

Additional statistics can be found in Figure 3.1 and Table 3.1. We use a paired t-test with $p \leq 0.05$ to assess statistically significant differences. We find that the random forest classifier outperforms both logistic regression and MLP Classifier across all evaluated metrics, while MLP Classifier also achieves a statistically significant improvement over logistic regression.

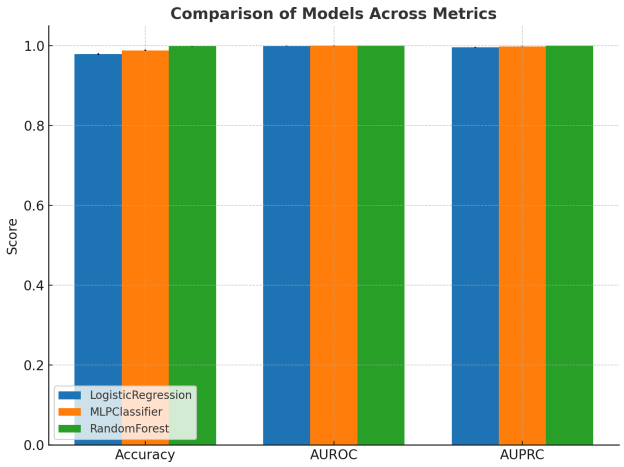

Figure 3.1 Grouped bar chart comparing the accuracy, AUROC, and AUPRC of the three models from Table 3.1.

These results validate the effectiveness of both models in static ASL gesture classification. Increasing the number of images per class improved model accuracy and stability, confirming the importance of sufficient training data.

To determine the confusion matrices, we train the models using a 3:1 data split and perform inference on the testing set. The results can be seen in Figure 3.1. Confusion matrix analysis revealed that most errors occurred between letters with similar visual structure, as shown in Figures 3.2, 3.3, and 3.4. Nevertheless, the majority of predictions were accurate, and the use of normalized inputs ensured consistent performance across runs.

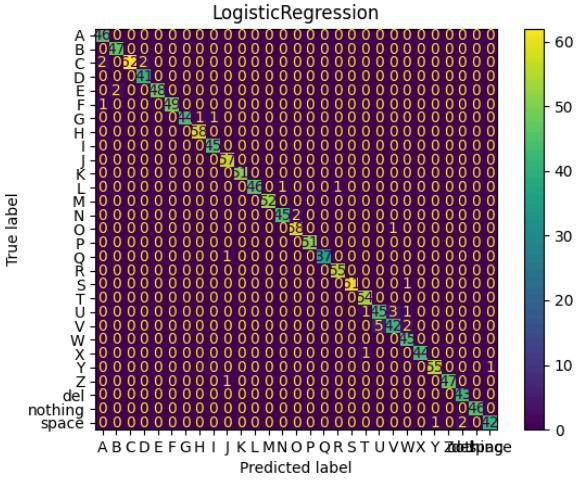

Figure 3.2. Confusion matrix for logistic regression.

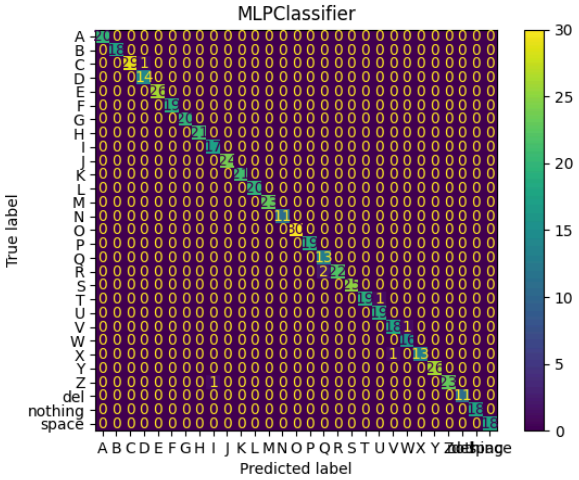

Figure 3.3. Confusion matrix for MLP classifier.

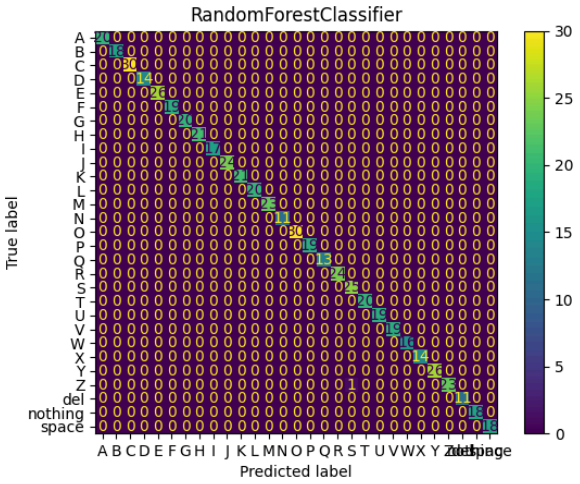

Figure 3.4. Confusion matrix for random forest classifier.

4. DISCUSSION

This study demonstrates the feasibility of using classical machine learning techniques for high-accuracy ASL alphabet recognition. The random forest classifier, in particular, displayed excellent generalization and flexibility, outperforming the MLP classifier in most experimental conditions.

We conjecture that the differences in model performance may be due to the differing capacity of the algorithms to capture non-linear feature interactions, with Random Forest's ensemble structure enabling more robust generalization compared to the shallower MLP, thereby explaining the statistically significant advantage observed.

If this were used in a real-time translation system, this system could be a promising tool for improving communication between individuals who use ASL and those who do not. Its deployment could assist in educational settings, healthcare, and customer service environments where real-time translation is beneficial.

Future Work: Enhancements to this model could include:

- Extending the system to dynamic sign sequences representing words or phrases.

- Augmenting the dataset to include variations in hand size, lighting, skin tone, and background clutter, as the current dataset did not have any variation.
- Upgrading the MLP and random forest models to CNNs, recurrent neural networks (RNNs), or vision transformers

5. CONCLUSION

The successful implementation of this system highlights how AI can support social inclusion. Real-world deployment on mobile or embedded platforms could bring tangible benefits to the hearing-impaired community and their interlocutors, especially when paired with user-friendly interfaces and dynamic gesture support. Continued work in this area holds the potential to make sign language more universally understood and accessible.

6. LIMITATIONS

Despite strong performance, the models exhibit limited generalizability. The dataset was captured under highly controlled conditions, raising concerns about overfitting and robustness in dynamic environments (Mohammadi et al., 2022). Moreover, both the MLP and Random Forest classifiers are shallow architectures restricted to fixed 64×64 grayscale inputs, constraining their ability to capture spatial hierarchies compared to CNNs (Pigou et al., 2015; Radford et al., 2021). Finally, the task is narrowly defined to static ASL letters, neglecting dynamic gestures and facial expressions essential to fluent communication. Temporal architectures such as RNNs or Transformers offer more suitable frameworks (Wolf et al., 2020).

REFERENCES

1. Akash Nagaraj. (2018). ASL Alphabet [Data set]. Kaggle. https://doi.org/10.34740/KAGGLE/DSV/29550
2. Bishop, C. M. (2007). Pattern recognition and machine learning. *Journal of Electronic Imaging*, *16*(4), 049901. https://doi.org/10.1117/1.2819119
3. Breiman, L. (2001). Random Forests. *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/a:1010933404324

4. Mohammadi, M., Chandarana, P., Seekings, J., Hendrix, S., & Zand, R. (2022). Static hand gesture recognition for American sign language using neuromorphic hardware. *Neuromorphic Computing and Engineering*, *2*(4), 044005. https://doi.org/10.1088/2634-4386/ac94f3
5. Pigou, L., Dieleman, S., Kindermans, P., & Schrauwen, B. (2015). Sign language recognition using convolutional neural networks. In *Lecture notes in computer science* (pp. 572–578). https://doi.org/10.1007/978-3-319-16178-5_40
6. Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., & Sutskever, I. (2021). Learning transferable visual models from natural language supervision. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2103.00020
7. Rokach, L., & Maimon, O. (2006). Decision Trees. In *Springer eBooks* (pp. 165–192). https://doi.org/10.1007/0-387-25465-x_9
8. Temitope, A., Nguyen Thanh, H., & Victor, A., 2025. Domain Knowledge in Feature Engineering: Why Human Intuition Still Matters. [online] ResearchGate. https://www.researchgate.net/publication/390492801_Domain_Knowledge_in_Feature_Engineering_Why_Human_Intuition_Still_Matters
9. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T.L., Gugger, S., Drame, M., Lhoest, Q. and Rush, A.M., 2020. *Transformers: State-of-the-art Natural Language Processing*. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pp.38-45. https://doi.org/10.18653/v1/2020.emnlp-demos.6