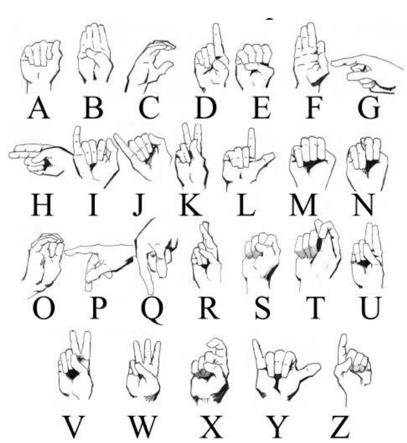
SignVision AI: A computer vision approach to visual translation and its risks

*

Mikayla Norton
M.S. Data Science
College of Natural Science
norton18@msu.edu

***Abstract*—SignVision AI focuses on the development of an advanced image classification system designed to accurately recognize sign language alphabets. Leveraging state-of-the-art machine learning algorithms and deep neural networks, the system interprets and classifies images of hand signs into the corresponding letters of the alphabet. This initiative serves as a crucial tool for facilitating communication within the deaf and hard-of-hearing community. By employing sophisticated image processing techniques and a comprehensive dataset of sign language alphabet gestures, the model achieves remarkable accuracy in real-time sign language recognition, enabling seamless communication and interaction.**

Addressing the integrity and reliability of such systems is critical, especially considering the potential for adversarial attacks. Adversarial attacks consist of subtly modified inputs that are engineered to deceive machine learning models into making incorrect predictions. The research further investigates the susceptibility of the sign language classification system to these attacks, showing how maliciously altered images could lead to the misinterpretation of sign language gestures. Through a series of experiments, SignVision AI illustrates the impact of various adversarial attack techniques on the model's performance and underscores the importance of robust defense mechanisms. By exploring these risks and implementing appropriate countermeasures, this endeavor not only contributes to the advancement of computer vision in sign language recognition but also ensures the security and reliability of such systems in practical applications, protecting against potential misuse.



I. Objectives

A. Primary Goal

The primary goal of this computer vision application is to develop a robust model capable of classifying American Sign Language (ASL) alphabet images with high accuracy and efficiency. By harnessing the potential of advanced machine learning and deep learning techniques, this model aims to facilitate seamless communication for the deaf and hard-of-hearing community through the accurate interpretation of sign language gestures. The development of such a system is not only a step forward in inclusive technology but also a significant contribution to the field of computer vision, pushing the boundaries of what is achievable in real-time gesture recognition.

B. Key Milestones

The first objective in achieving this goal involves the initial development of the base model for ASL alphabet classification. This phase focuses on designing and training a deep neural network that can accurately recognize and classify a wide range of sign language gestures into their corresponding alphabet letters. The processes include curating a comprehensive and diverse dataset of ASL images, selecting an appropriate model architecture, and employing advanced image processing techniques to enhance model training and performance.

Upon establishing a baseline model, the next milestone centers on assessing and demonstrating the model's vulnerability to adversarial attacks. This involves crafting and deploying various adversarial attack strategies to subtly modify ASL images in a way that deceives the model into making incorrect classifications. By exposing the model's weaknesses, this phase aims to highlight the critical need for robustness in machine learning systems, especially those used in sensitive and impactful applications such as sign language recognition.

The final step in this project is the development of an improved, robust version of the initial model that is resistant to adversarial attacks. Achieving this involves implementing advanced defensive mechanisms, such as adversarial training, where the model is trained on both original and adversarially modified images to improve its resilience. Additional techniques may include regularization methods and the exploration of novel neural network architectures designed to enhance model robustness. This phase aims to deliver a sign language classification system that not only performs with high accuracy under normal conditions but also maintains its reliability and integrity in the face of adversarial challenges, ensuring its practical applicability in real-world scenarios.

II. Background and Motivation

A. Problem Statement

This initiative centers on the development of a robust computer vision model capable of accurately classifying American Sign Language (ASL) alphabet images, aiming to achieve precise gesture recognition for seamless communication within the deaf and hard-of-hearing community. Additionally, it addresses the critical need for the model's defense against adversarial attacks, which threaten its integrity by manipulating input data to elicit incorrect predictions.

B. Importance of Problem

By enabling precise and efficient gesture recognition, such a model directly supports the communication needs of the deaf and hard-of-hearing community. It bridges a crucial gap in interactive technology, making digital platforms and services more accessible and inclusive. This advancement fosters equality in communication, education, and access to information, which are fundamental rights.

The emphasis on defending against adversarial attacks addresses a growing concern in the field of artificial intelligence (AI) and machine learning: the security and reliability of AI systems. Adversarial attacks, by their nature, can compromise the integrity of AI models, leading to erroneous outcomes that could have serious implications, especially in sensitive applications like sign language interpretation. By focusing on creating a system that is both accurate and resilient to such attacks, the initiative highlights the importance of trustworthiness and reliability in AI technologies.

This problem is emblematic of broader challenges in AI and machine learning regarding data integrity, model robustness, and the ethical implications of technology deployment in real-world scenarios. It underlines the need for ongoing research and innovation to ensure that AI systems not only perform their intended functions under normal conditions but also maintain their performance in adversarial or otherwise challenging environments. Thus, addressing this problem not only has immediate benefits for specific user groups but also contributes to the foundational integrity and ethical development of AI technologies at large [1].

III. Methodology

A. Data Collection

1) Data Sources: For the development of the ASL alphabet classification model, the training data was sourced from a comprehensive dataset available on Kaggle [2], which consists of high-quality images of hand signs set against a uniform white background. This dataset provides a wide range of gestures representing the ASL alphabet, ensuring diversity and coverage necessary for effective model training.

To evaluate the model's performance and its applicability in real-world scenarios, the testing data was manually generated. This involved capturing photographs of various hand signs performed by the researcher, mimicking the ASL alphabet gestures, against a similar white background. This approach to sourcing and generating data ensures that the model is trained on standardized inputs while being tested on data that mimics practical use cases, striking a balance between controlled training conditions and the variability encountered in everyday applications.

B. Initial Data Analysis

In the initial data analysis, the dataset was observed as meticulously organized into 29 distinct classes to comprehensively represent the ASL alphabet alongside crucial functional gestures. This classification includes the 26 standard letters of the alphabet, as well as specific images designated to represent "space" and "delete" commands in ASL, adding functionality to the gestural communication. Additionally, "blank" images, consisting solely of the uniform white background without any hand signs, were included to enhance the model's ability to recognize non-gestural frames. This diverse set of classes ensures the model's capability to interpret a wide range of inputs accurately. Distributed evenly across the dataset, each class is represented by 3,000 images, culminating in a total of 87,000 training images. This balanced distribution is crucial for preventing bias towards more frequently represented classes and ensures that the model develops a comprehensive understanding of the ASL alphabet and functional gestures, providing a solid foundation for accurate classification and recognition.

C. Exploratory Data Analysis

The Exploratory Data Analysis of the ASL alphabet dataset involved a systematic breakdown to comprehend the inherent structure and diversity of the image data. Given the high-dimensional nature of image data, Principal Component Analysis (PCA) was applied to reduce dimensionality while preserving the essence of the data. This reduction allows for a more manageable number of variables, or principal components, which encapsulate the most informative features of the original dataset.

The exploratory data analysis began with the application of principal component analysis on the standardized dataset, which revealed the amount of variance each principal component held. The scree plot generated from the PCA analysis — a graphical representation of the cumulative explained variance against the number of components — showed a swift ascent in explained variance within the first few components, followed by a plateau and is visualized in Figure 1 below.

This inflection point in Figure 1 suggests that beyond a certain number of components, the incremental gain in information becomes marginal. Consequently, this insight directs the dimensionality reduction where components beyond the identified number contribute minimally to the predictive power of the model.

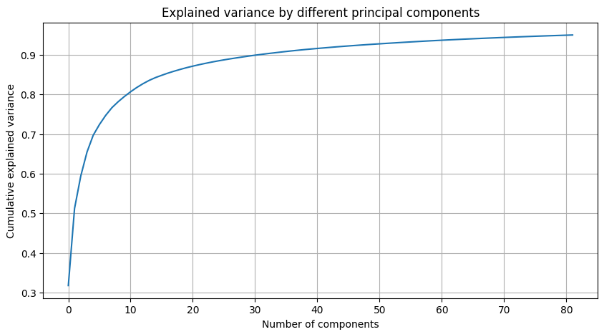


Fig. 1. Principal Component Analysis

D. Data Preparation

Prior to the PCA application, the data preparation process involved standardizing the dataset to have a mean of zero and a variance of one. This step, performed using scikit-learn's StandardScaler, is crucial to ensure that each feature contributes equally to the analysis and subsequent model training. The training set, composed of 87,000 images, was flattened into a two-dimensional array where each image was represented as a single row of pixels. This reshaping allowed the standardized pixel values to serve as individual features in the PCA.

Following standardization, the PCA was fitted to the training data, specifying that the number of components retained should explain 95% of the variance. This parameter choice is a trade-off between computational efficiency and data representation fidelity. The transformation resulted in a new data set with reduced dimensions, upon which the model could be trained more efficiently without a significant loss of information.

The labels for the images were concurrently processed to fit the model's requirements. Given the categorical nature of the classification task — with 29 distinct classes including the ASL letters, "space," "delete," and "blank" — the labels were converted into a binary matrix representation using a one-hot encoding scheme. This conversion was essential for the classification model to interpret the labels correctly during training and performance evaluation.

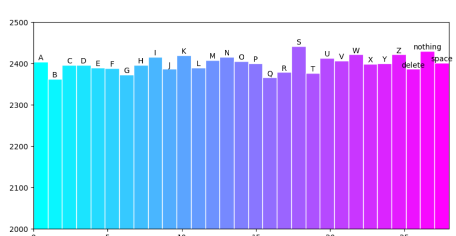


Fig. 2. Training Data Class Distribution

The data is then separated into training and validation sets, separate from the manually constructed testing set. The training set is distributed into the 29 classes as shown in Figure 2.

E. Baseline Model Development

1) Architecture: The model architecture was constructed by appending a custom top layer to a pre-existing base model, which was chosen for its efficacy in feature extraction from image data. The top layer comprises a dense layer with an output size equal to the number of classes, using softmax activation to facilitate multi-class classification. This architecture effectively integrates the rich feature-detection capabilities of the base model with a tailored top layer suited to the specifics of the ASL dataset.

An initial baseline model training used a computationally expensive tactic of using unscaled and full-dimension data, which demonstrated 91% training accuracy. The second baseline model followed the above procedure for flattened data, training significantly quicker, but a increasing to a higher level of training accuracy at 97%. However, further exploration revealed that while the training and validation accuracies were high, the testing accuracy was quite low ($\approx 7\%$), indicating an overfit of the model.

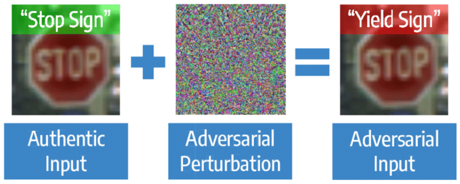
A new adaptation was taken to the model approach to generalize the target training. The new single dimensional model fit was computationally inexpensive with a higher testing accuracy, however the training and validation accuracy decreased in tradeoff.

2) Training: Training the model involved a two-step process. Initially, all layers of the base model were frozen to maintain their pre-trained weights, focusing the training exclusively on the newly added top layer. The model was compiled with categorical cross-entropy loss and a specified optimizer. Training utilized early stopping based on validation accuracy to prevent overfitting. The process involved iterating over the scaled and PCA-transformed dataset for a set number of epochs, adjusting for class imbalances with class weights where necessary.

3) Evaluation and Tuning: Post-training, the model's performance was evaluated using the hold-out test set that had also been transformed by PCA and scaling. The evaluation criteria included accuracy and loss metrics to gauge the model's generalization capability. Based on the initial performance, the model underwent fine-tuning, which included unfreezing select layers of the base model for additional training, modifying the learning rate, and adjusting epoch count, if necessary. This iterative process of evaluation and tuning aimed to optimize the model's predictive accuracy on unseen data.

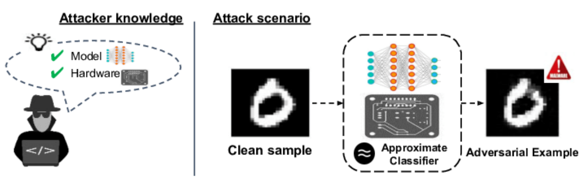
F. Adversarial Attacks

An adversarial attack in the context of machine learning refers to a technique by which an attacker makes subtle, often imperceptible changes to input data with the intent to deceive a machine learning model into making incorrect predictions or classifications [3]. These manipulated inputs, known as adversarial examples, exploit weaknesses or blind spots in the model's understanding of the input space. While to a human observer these perturbations might go unnoticed, they can cause the model to misinterpret the data and produce erroneous outputs.



This vulnerability poses significant risks, especially in critical applications such as security systems, autonomous vehicles, fraud detection, and translations. Adversarial attacks highlight the importance of considering robustness and security in the design and deployment of machine learning systems, prompting the development of defensive strategies to counteract such manipulation.

A white-box adversarial attack is a strategy where the attacker has complete knowledge of the model's architecture, parameters, and training data. This transparency enables the attacker to craft perturbations tailored to the model's specific weaknesses. In the context of image classification, such an attack manipulates input images with imperceptible changes that are designed to confuse the model into making incorrect predictions [4].



Applied to the ASL classifier, a white-box adversarial attack would involve generating manipulated versions of the ASL alphabet images from the testing set. These images would be altered in such a subtle way that, while they would appear unchanged to the human eye, they would be significantly different for the classifier, leading to misclassification. The attacker would use gradients of the model (since they have full knowledge of it) to determine the most effective perturbations for each image. Using Foolbox [5], a Python library for generating adversarial attacks, an adversarial attack will be demonstrated on the constructed baseline model to showcase the dangers of these attacks.

In response to the adversarial attack, the baseline model will be retrained using these adversarially altered images alongside the original training set in a process known as adversarial training. By exposing the model to both legitimate and adversarial samples during retraining, the model will learn to recognize and resist these perturbations, enhancing its robustness. This retraining process would ideally lead to a model that maintains high classification accuracy in the presence of adversarial inputs, ultimately fortifying the system against potential attacks.

G. Metrics for Success

The success of the baseline and robust models were measured using several key performance metrics:

1) Accuracy: This metric represents the proportion of correctly classified instances out of all predictions and provides a quick snapshot of model performance. It's especially useful as an initial indicator of success.

2) Classification Report: The classification report breaks down the performance by class, providing precision, recall, and F1-score for each label. Precision indicates the accuracy of positive predictions for each sign, recall measures the model's ability to detect all relevant instances of a class, and the F1-score provides a balance between precision and recall, which is crucial for models where an equilibrium between the two is desired.

3) Confusion Matrix: The confusion matrix offers a detailed visual and numeric representation of the model's performance, showcasing the correct and incorrect predictions across all classes. It helps in identifying classes that are commonly misclassified, highlighting potential areas for model improvement.

IV. Communication of Results

A. Reporting

Much like this proposal, detailed documentation has been and will continue to be collected through the duration of this project to successfully capture the methodology and performance of various models created. Specific highlights of the future developments to be reported include:

1) Adversarial Attack (White Box Attack) Results:

- Attack Success Rate: Document the success rate of the white box attack in misleading the baseline model.
- Impact on Metrics: Discuss how the adversarial attack affected the baseline model's accuracy and other metrics.
- Visual Examples: Provide side-by-side visual examples of original and adversarially altered images to demonstrate the changes.

2) Defensive Model Development:

- Enhancements Introduced: Describe the enhancements and modifications made to the baseline model to defend against adversarial attacks.
- Performance Comparison: Compare the performance of the defensively enhanced model with the baseline model post-attack.
- Robustness Metrics: Report on specific robustness metrics, such as adversarial accuracy or attack detection rates.

B. Visualization

The final delivery of this study on ASL alphabet image classification will be encapsulated in an interactive Streamlit web application, which will serve as both an informative and demonstrative platform for users to engage with the research findings and its practical applications.

1) Product Homepage: The homepage will introduce visitors to the purpose and utility of the application, providing a succinct overview of the project's objectives and the significance of the models developed. It will guide users on navigating the site and utilizing the tools and demonstrations available.

2) Baseline Model Demonstration: A dedicated subpage will showcase the baseline model's capabilities, displaying its classification performance on the testing data. Users will have the opportunity to view the model's predictions and the corresponding accuracy metrics in an interactive format.

3) Adversarial Attack Background: This informational subpage will educate users on the concept of adversarial attacks, specifically the white-box approach utilized in this study. It will lay the groundwork for understanding the vulnerability of machine learning models to such attacks and the importance of robust defenses.

4) White Box Attack Demo: An interactive demonstration will allow users to witness the effect of a white-box attack first-hand. The page will display adversarial samples next to their originals, visually highlighting the perturbations and the model's altered predictions.

5) Improved Model Demo: Following the demonstration of the white-box attack, this subpage will introduce the improved model that has been retrained to resist such adversarial tactics. Users will be able to test the enhanced model's accuracy and its defense against adversarial samples, offering a clear before-and-after comparison.

6) Results/Methods in Detail: For those seeking an in-depth understanding of the research, a comprehensive subpage will detail the methodology, results, and analyses. This section will include all pertinent data, visualizations, and discussions, offering a deep dive into the study's scientific underpinnings.

7) Biography Page: The biography page will provide information about the contributors to the project, acknowledging their roles and expertise. This personal touch adds credibility and allows users to connect with the team behind the research.

In sum, the Streamlit application will serve as a dynamic and user-friendly means of delivering the project's findings, demonstrations, and implications, catering to both laypersons and experts interested in the field of ASL image classification and adversarial machine learning.

V. Challenges

A. Potential Risks and Mitigation

Scalability and Performance

R: The web application might experience performance issues under high traffic.

M: In future development, use cloud-based services with auto-scaling capabilities and optimize the model for high performance, ensuring swift response times even under increased load.

Diversity and Inclusivity

R: The model might exhibit biased performance due to a lack of diversity in the training dataset.

M: Find new datasets to train the model with a diversity that represents different hand sizes, skin tones, and signing styles.

M: Offer a feedback mechanism to capture and integrate user-reported misclassifications.

Legal and Ethical Considerations

R: Misinterpretations by the ASL model could lead to miscommunication and potential legal issues.

M: Provide clear disclaimers regarding the limitations of the model.

B. Ethical Considerations

The ethical landscape of employing AI for ASL recognition necessitates careful consideration. It's crucial to ensure that the technology does not replace human sign language interpreters in critical scenarios where nuances and emotional contexts are essential. Transparency in how the AI model makes decisions and handles errors is also vital to maintain user trust.

Furthermore, the creation and deployment of adversarial examples must be handled responsibly, ensuring that the knowledge and tools for generating such examples do not fall into malicious hands. There must be a commitment to ethical AI practices, including responsible disclosure of vulnerabilities and collaborative efforts to improve the security of machine learning models across the industry.

By acknowledging and addressing these potential risks and ethical considerations, the project can ensure a responsible and conscientious approach to developing and deploying ASL classification models.

VI. Preliminary Results

As previously indicated in earlier sections, the project has progressed through initial stages of baseline model development and Foolbox attack practice on sample models.

The results of the current baseline model are elaborated as follows:

- On a training set of 80% of the initial data and validation set of the remaining 20%, the current accuracies are 65% and 82% respectively for training and validation. The classification report is displayed in Table 1 of the appendix.
- To identify trends of possible sign confusion, a confusion matrix was generated to display commonly mistaken signs for the validation set. Results are shown in Table 2 of the appendix.
- On sample substitute training from the baseline model white-box attack, currently the attack creates a 90% or above misclassification rate.
- Upon the implementation of adversarial defense, the goal is to minimize the adversarial attack success to a margin of $\pm 5\%$ of the baseline testing accuracy.

VII. References

[1] N. Papernot, P. D. McDaniel, I. J. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, ”Practical Black-Box Attacks against Deep Learning Systems using Adversarial Examples,” CoRR, vol. abs/1602.02697, 2016. [Online]. Available: http://arxiv.org/abs/1602.02697

[2] A. Nagaraj, ”ASL Alphabet,” Kaggle, 2018. [Online]. Available: https://www.kaggle.com/dsv/29550, DOI: 10.34740/KAGGLE/DSV/29550.

[3] J. Gomes, ”Adversarial Attacks and Defenses for Convolutional Neural Networks,” Onfido Product and Tech, Jan. 16, 2018. [Online]. Available: https://medium.com/onfido-tech/adversarial-attacks-and-defences-for-convolutional-neural-networks-66915ece52e7.

[4] W. Schroeder, ”Learning Machine Learning Part 2: Attacking White Box Models,” Posts By SpecterOps Team Members, Apr. 26, 2022. [Online]. Available: https://posts.specterops.io/learning-machine-learning-part-2-attacking-white-box-models-1a10bbb4a2ae.

[5] J. Rauber and R. S. Zimmermann, ”Foolbox 3.3.3 Documentation,” 2021. [Online]. Available: https://foolbox.readthedocs.io/

VIII. Appendix

TABLE I
Classification report for the ASL alphabet baseline recognition model.

Class	Precision	Recall	F1-Score	Support
A	0.80	0.76	0.78	614
B	0.81	0.80	0.80	605
C	0.82	0.86	0.84	603
D	0.78	0.67	0.72	561
E	0.76	0.82	0.79	603
F	0.92	0.78	0.85	582
G	0.90	0.76	0.82	615
H	0.92	0.86	0.89	590
I	0.97	0.76	0.85	584
J	0.80	0.94	0.86	587
K	0.71	0.96	0.82	606
L	0.90	0.80	0.85	570
M	0.81	0.82	0.82	630
N	0.98	0.90	0.94	604
O	0.68	0.88	0.77	592
P	0.93	0.90	0.91	540
Q	0.89	0.97	0.93	577
R	0.86	0.60	0.71	624
S	0.69	0.92	0.79	611
T	0.80	0.79	0.79	626
U	0.63	0.65	0.64	611
V	0.61	0.75	0.67	605
W	0.79	0.69	0.73	603
X	0.86	0.86	0.86	599
Y	0.77	0.78	0.78	605
Z	0.97	0.87	0.92	565
delete	0.93	0.84	0.88	541
nothing	1.00	0.96	0.98	617
space	0.90	0.86	0.88	631
accuracy			0.82	17400
macro avg	0.83	0.82	0.82	17400
weighted avg	0.83	0.82	0.82	17400

TABLE II
Confusion matrix of validation data for ASL alphabet baseline recognition model.