Project 3 Applied Data Science

Abstract

This project explores the translation of American Sign Language (ASL) fingerspelling gestures into English letters using a multimodal approach that combines RGB and depth data. Building on existing idea on sign language recognition using a multimodal deep learning approach, the project leverages a convolutional neural network (CNN) architecture that incorporates both visual and spatial data for improved gesture recognition [1]. This project aim to explore when and how much including depth image improves the ASL recognition. Despite challenges with dataset quality and training time, the model demonstrates promising results, with RGB images achieving an accuracy of 85.86% on the test set, while depth data alone performs much worse. The combination of both modalities yields a modest improvement of around 60%, with an improving behaviour for recognising ‘m’ comparing with pure RGB image. This study provides insights into the potential and limitations of multimodal systems for ASL translation, highlighting the need for further testing and model refinement to achieve real-world applicability.

Introduction

American Sign Language (ASL) is a crucial communication tool for the Deaf and Hard of Hearing communities, but language barriers remain a significant challenge between ASL users and non-signers. Traditional ASL translation systems often rely solely on RGB images, which can miss important spatial information in hand gestures. Motivated by the potential of multimodal learning, as demonstrated by Amutha et al. [1], this project combines RGB and depth data to enhance the translation of ASL fingerspelling gestures into English letters. By using depth information alongside RGB images, the aim is to improve gesture recognition accuracy, accounting for challenges such as hand shape variations, hand directions, and environmental noise. The research seeks to explore the effectiveness of multimodal fusion in ASL translation systems and contribute to the development of more accessible, real-time tools for communication.

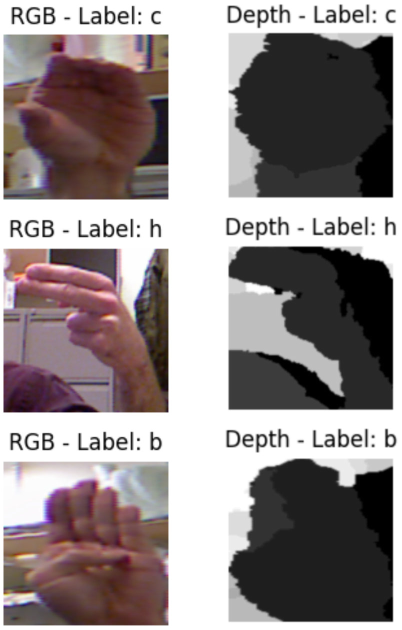


Fig 1. Mapped RGB image and Depth image.

Methods

Data Preprocessing

- **Data Mapping**: The original dataset consists of unpaired RGB and depth images for each ASL fingerspelling gesture except j and z from 5 people (5 sets), all .png images with different sizes, with labels corresponding to the letters of the alphabet [2]. The preprocessing of data paired them up based on the labels. The filenames were analyzed to pair the corresponding RGB and depth images for each sample. As shown in **Fig 1**.
- **Label Encoding**: The ASL letters were encoded into numerical labels using scikit-learn’s `LabelBinarizer`, which transformed the categorical labels (A-Z) into a binary

matrix for multi-class classification.

- **Image Resizing**: To standardize the inputs across the model, both RGB and depth images were initially resized to 224x224 pixels. The images were then normalized to a range of [0, 1] by dividing pixel values by 255.0.
- **Data Augmentation:** For improving model generalization and reducing overfitting, augmentation techniques were applied, including random rotations, flips, and slight translations of both RGB and depth images.
- **Data Splitting**: The dataset was split into training (60%), validation (20%), and test (20%) sets, no overlap or dependence between each set, all are real-world hand ASL images with messy background as where ASL translation would be needed. Divided sampling was used to ensure each letter was proportionally represented in each subset.

Feature Extraction & Scaling

- **Feature Extraction:** The model architecture leverages pre-trained ResNet50 for both RGB and depth image feature extraction. The output of the ResNet50 model, after removing the connected layers, is passed through a global average pooling layer to extract the most relevant features from both RGB and depth images.
- **Scaling**: RGB and depth images were normalized and scaled to a range of [0, 1] to aid convergence during model training. This preprocessing ensured the model could learn efficiently and avoid biases from different image intensities.

Multimodal Fusion & Model Architecture

- **Multimodal Data**: A custom data generator was used to load paired RGB and depth images. This generator ensures that both image types are processed simultaneously during training, enabling multimodal fusion. The generator resizes, normalizes, and augments the images while reducing memory overhead during training.
- **Multimodal Fusion Model**: A multimodal model was designed using two separate ResNet50 networks: one for RGB images and the other for depth images. The feature maps extracted from both networks were concatenated and passed through several connected layers. This architecture aimed to leverage visual and spatial information from the images for improved recognition.
- **Custom Layers**: The features were passed through two dense layers with ReLU activations and dropout for regularization. The final output layer used a softmax activation.

Models

- **VGG19**: VGG19 model was used for RGB and depth image processing seperately. This pre-trained model was fine-tuned for ASL fingerspelling translation. VGG19 for only RGB image was regarded as the baseline, taking it as closer to traditional ASL translation systems.
- **Multimodal ResNet50**: Both the RGB and depth networks were fused, aimed to improve the gesture recognition by incorporating both the appearance and spatial aspects of the images.

Evaluation Metrics

- **Accuracy**: The primary metric to compare models.
- **Confusion Matrix**: To analyze misclassification patterns across ASL letters.

Code Availability:

GitHub https://github.com/EvelynnnnnnZ/Applied-Data-Science-Fall-2024

Results

Model Performance

- **VGG19-RGB**: Achieved the highest accuracy above 85%, outperforming other models due to its ability to handle small datasets with imbalanced classes. This model demonstrated strong performance with a clear learning behavior, especially given the limited dataset. It was able to learn meaningful features from the images, as indicated by its ability to correctly classify the dominant ASL letters with high accuracy and pattern tend to converge, as shown in **Fig 2.** To enhance interpretability, case-level true vs. predicted labels were examined, providing insights into where the model struggles. As shown in **Fig 3,** the model's predictions were generally correct, with notable success for more common ASL letters in the dataset, while getting confused on 'm', 'n', and 'e', which are more 'similar' in gesture. In **Fig 4**, VGG19-RGB model is still making reliable recognition for ASL letters with unseen image.

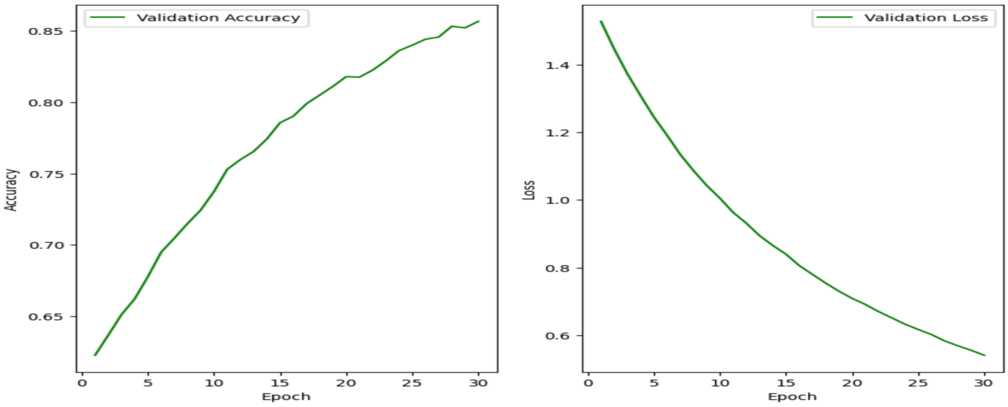


Fig 2. Validation Accuracy and loss for VGG19-RGB.

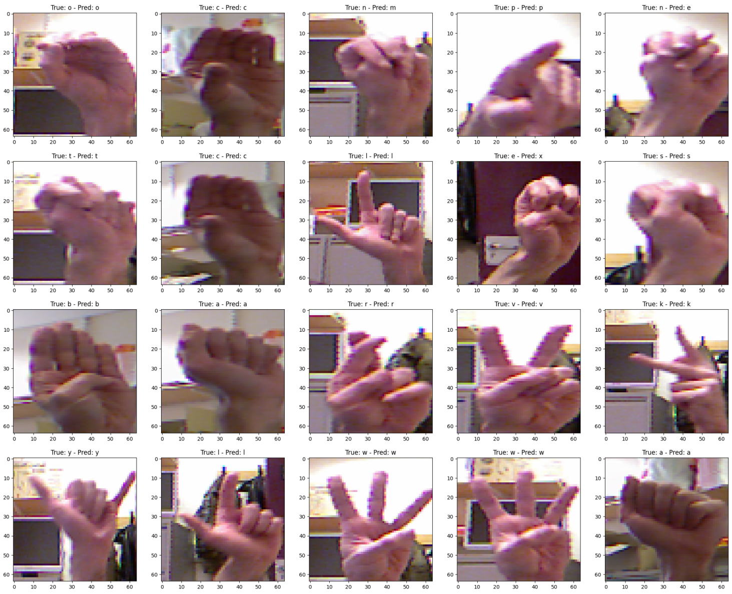


Fig 3. Prediction matching with true letter of VGG19-RGB.

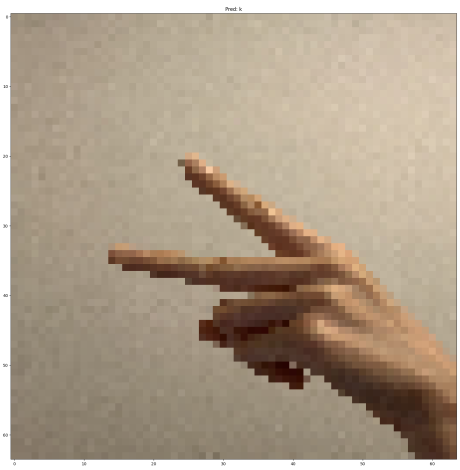


Fig 4. VGG19-RGB makes correct recogniti on with unseen image.

- **VGG19-Depth**: The VGG19 with Depth image performed not very well comparing with other models. It starts with a very low accuracy being 5%, possibly indicate the ability on recognizing ASL letters with just depth image is relatively poor.
- **Multimodal Model**: The multimodal model, which combined both RGB and depth images, achieved an accuracy of above **60%**. While this represented an improvement over the depth-only model, it still underperformed compared to the RGB model. The confusion matrix for the multimodal model, as in Fig 5, showed a reasonable correctness in classification, but still includes some off-diagonal values indicating misclassifications. This suggests that the multimodal approach did not fully leverage the complementary nature of the two data types, possibly indicate the need of different custom CNN. However, comparing with VGG19-RGB, multimodal model have a better behaviour on ‘m’, which shows that it has potential on distinguishing similar ASL gestures.

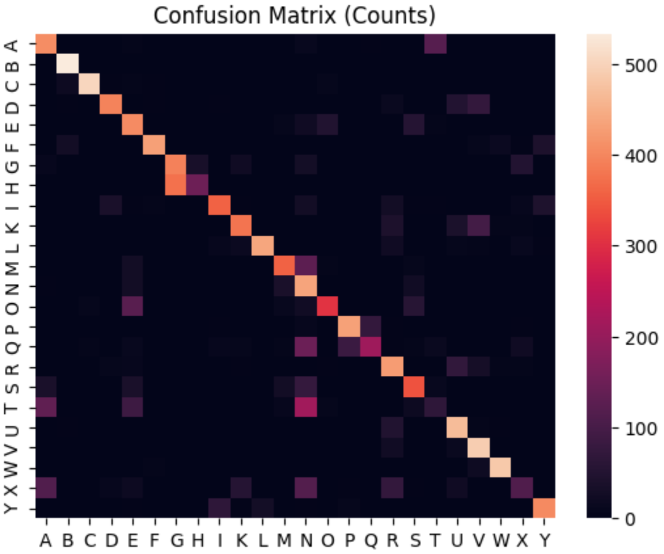


Fig 5. Confusion matrix of multimodal model.

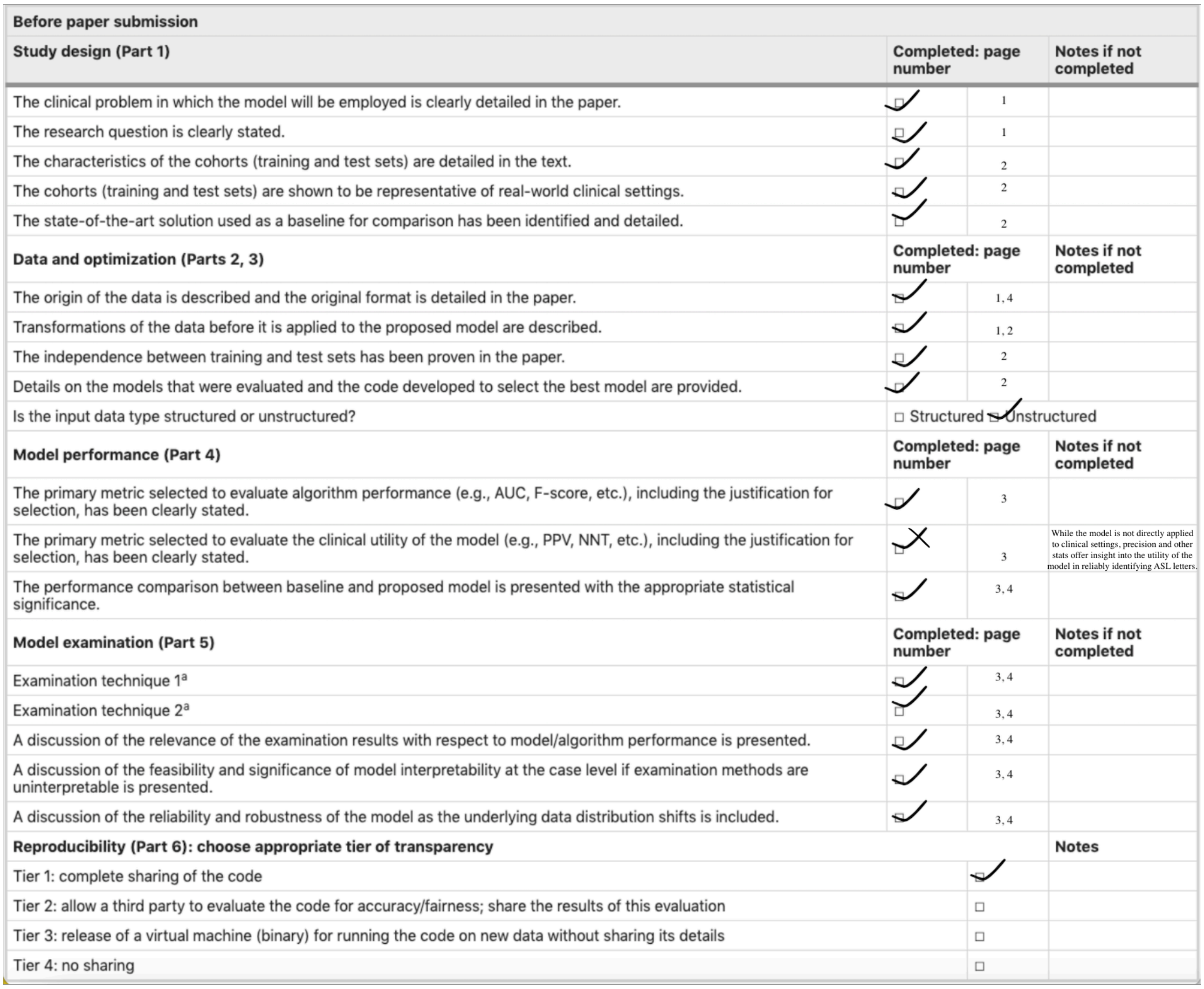
Conclusion

This project investigated the use of multimodal learning for translating American Sign Language (ASL) fingerspelling gestures into English letters by combining RGB and depth images. The VGG19-RGB model achieved the highest accuracy of 85.86%, while the depth-only model performed poorly, and the multimodal model combining both inputs showed moderate behavior overall, with more potential in distinguishing ASL letters with similar gesture as some other letters, like ‘m’. The results highlight the robustness of RGB data for hand gesture recognition and the challenges and potential of effectively integrating depth information.

Future work should work on more complex and adjusted multimodal model, expand the dataset to include more sets of images, and explore more advanced fusion techniques to fully leverage the complementary nature of RGB and depth data. Including larger dataset for letters still remains hard to distinguish, like ’n’ and ‘e’, would also possibly bring improvements. Training with GPU resources and work to include dynamic gestures ‘j’ and ‘z’ could improve efficiency and generalization. This project provides a foundation for developing real-time ASL-to-English translation systems to enhance communication accessibility for Deaf and Hard of Hearing communities.

Reference

[1] Amutha, S., et al. (2023). Real-Time Sign Language Recognition using a Multimodal Deep Learning Approach. *IEEE Access*, 11, 81356-81365. https://doi.org/10.1109/ACCESS.2023.10199569
[2] Geislinger, V. ASL RGB-Depth Fingerspelling Dataset. Kaggle, 2022. https://www.kaggle.com/datasets/mrgeislinger/asl-rgb-depth-fingerspelling-spelling-it-out/data?select=dataset5



Before paper submission			
Study design (Part 1)	**Completed: page number**		**Notes if not completed**
The clinical problem in which the model will be employed is clearly detailed in the paper.	☑	1	
The research question is clearly stated.	☑	1	
The characteristics of the cohorts (training and test sets) are detailed in the text.	☑	2	
The cohorts (training and test sets) are shown to be representative of real-world clinical settings.	☑	2	
The state-of-the-art solution used as a baseline for comparison has been identified and detailed.	☑	2	
Data and optimization (Parts 2, 3)	**Completed: page number**		**Notes if not completed**
The origin of the data is described and the original format is detailed in the paper.	☑	1, 4	
Transformations of the data before it is applied to the proposed model are described.	☑	1, 2	
The independence between training and test sets has been proven in the paper.	☑	2	
Details on the models that were evaluated and the code developed to select the best model are provided.	☑	2	
Is the input data type structured or unstructured?	□ Structured ☑ Unstructured		
Model performance (Part 4)	**Completed: page number**		**Notes if not completed**
The primary metric selected to evaluate algorithm performance (e.g., AUC, F-score, etc.), including the justification for selection, has been clearly stated.	☑	3	
The primary metric selected to evaluate the clinical utility of the model (e.g., PPV, NNT, etc.), including the justification for selection, has been clearly stated.	☒	3	While the model is not directly applied to clinical settings, precision and other stats offer insight into the utility of the model in reliably identifying ASL letters.
The performance comparison between baseline and proposed model is presented with the appropriate statistical significance.	☑	3, 4	
Model examination (Part 5)	**Completed: page number**		**Notes if not completed**
Examination technique 1[a]	☑	3, 4	
Examination technique 2[a]	☑	3, 4	
A discussion of the relevance of the examination results with respect to model/algorithm performance is presented.	☑	3, 4	
A discussion of the feasibility and significance of model interpretability at the case level if examination methods are uninterpretable is presented.	☑	3, 4	
A discussion of the reliability and robustness of the model as the underlying data distribution shifts is included.	☑	3, 4	
Reproducibility (Part 6): choose appropriate tier of transparency			**Notes**
Tier 1: complete sharing of the code		☑	
Tier 2: allow a third party to evaluate the code for accuracy/fairness; share the results of this evaluation		□	
Tier 3: release of a virtual machine (binary) for running the code on new data without sharing its details		□	
Tier 4: no sharing		□	