American Sign Language Final Project Report (Milestone 2)
Group #5

I. Basic Project Information

1. List of team-members and main tasks accomplished:

- **Alex Makienkov**
  - Identifying, downloading, and assembling datasets from various sources into an accessible database.
- **Alon Mecilati**
  - Preprocessing images by converting them into landmark-data using *mediapipe* library; creating and organizing GitHub repo.
- **Ariel Drabkin**
  - Performing EDA; creating first-draft of EDA report.
- **Yonatan Abrams**
  - Editing EDA report; Writing and editting final report submission.

2. Project Name and Achievements:

- For now, the project name is '*ASL SpellNet*'.
  - **GitHub Repo**: https://github.com/Altonormz/ASL_SpellNet

- "*State of the project*":
  - Assembled a robust dataset of ASL images of hands, documenting the process for this assembly, including download instructions with links. Transformed all images into landmark data in 3-dimensions, performed exploratory data analysis on the dataset, preprocessed the data in some minor ways to facilitate the exploration and wrote a summary of findings.
  - Trained a simple Random Forest Classifier model to identify the images, with an average accuracy of ~95% for each label respectively.

3. Business uses of this project:

Developing a machine learning model for sign language decoding can open a multitude of business applications in a variety of contexts/industries. Here are several examples:

- Lifestyle: Facilitating deaf people to communicate in real time with someone without ASL experience even without hiring an interpreter, which can be debilitatingly costly to hire for simple social interactions, and also trivializes the intimacy, privacy and fulfillment that personal friendships can offer.

- Healthcare: Avoiding miscommunication between patients and medical personnel using an "ASL-To-Text" application which the "disabled" (henceforth referred to as "differently abled") person can corroborate in real time.

- Telecommunication: Platforms such as Zoom, Slack's Huddle and other teleconferencing solutions can employ an "ASL-To-Text" model to deliver ASL interpretation, increasing accessibility and furthering inclusion in both business and social contexts.

- Smart Services: Increasing accessibility to services like Alexa or Siri so those without the capability to verbalize may also engage with these powerful tools, increasing accessibility and furthering inclusion.

II. Exploratory Data Analysis (EDA)

1. The Data Sets:

1.a. Summary of main sources for Data Sets comprising the complete dataset

1. ASL Alphabet (Images) (87,000 images):
    - https://www.kaggle.com/datasets/grassknoted/asl-alphabet
    - The test set for this dataset is from Kaggle too:
        - https://www.kaggle.com/datasets/danrasband/asl-alphabet-test

2. ASL Fingerspelling (Images) (2.34 GB):
    - https://www.kaggle.com/datasets/mrgeislinger/asl-rgb-depth-fingerspelling-spelling-it-out

3. Word Level ASL (Video) (5.4 GB):
    - https://www.kaggle.com/datasets/risangbaskoro/wlasl-processed

4. ASL dataset compiled from Github and Kaggle by Alex Makienkov:
    - https://mega.nz/file/8vUxiYQD#Lz9bflZQ-7-ESiGRcJ38gQto091oAkGfsdANGnKKVqI

1.b Description of complete dataset used for this EDA

For our analysis, we have gathered a dataset consisting of 428,644 images from several Kaggle contests depicting the 26 alphabet letters of the American Sign Language, along with images representing the "delete" and "space" signs. Each image was processed to extract the landmarks corresponding to the hand shape for the respective letter.

The hand landmarks data provides us with the x-y-z coordinates for each key-point in each hand. There are a total of 21 hand landmarks, and each landmark is defined by its x, y, and z coordinates. To ensure consistency, the x and y coordinates have been normalized to a range of [0.0, 1.0] using the width and height of the image, respectively.

The z coordinate represents the depth of the landmark, with the origin set at the wrist. A smaller z value indicates that the landmark is closer to the camera. The magnitude of the z coordinate follows a similar scale to the x coordinate.

By analyzing these coordinates, we can gain insights into the hand shape and its variations for each letter in the American Sign Language.

2. “Landmark” scheme:

To analyze this dataset, it is important to understand the mapping used for identifying and labelling landmarks from an image of a hand. There are 21 landmarks used in this analysis, as identified in *Figure 1*.

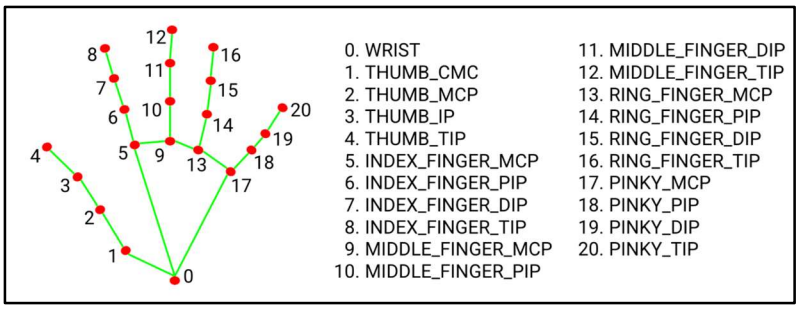

Figure 1 - Reference: https://developers.google.com/mediapipe/solutions/vision/hand_landmarker

3. Size, Shape, and Distribution of Dataset:

After combining all the images together, the following were the dimensions of the dataset:

- 428,644 samples/rows
- 65 columns:
    - **col 1: "Image_name"** (the filename which was the source of the sample).
    - **cols 2-64:** *[21 landmarks for each of three dimensions (x, y, and z)]*
    - **col 65: "letter"** *[letter/"label"]*

3.a. Example samples

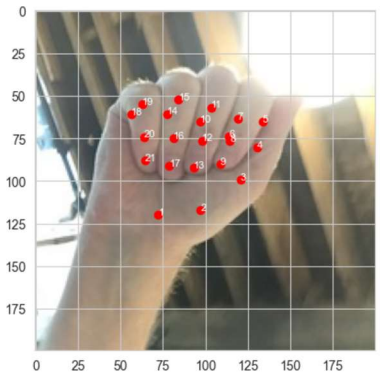

Figure 2 - The Letter "a". In this visualization, we have aligned the extracted landmark coordinates on the actual image of the letter 'a'. By overlaying the landmarks on the image, we can directly observe the correspondence between the spatial coordinates and their visual representation. This alignment provides a visual confirmation of the accuracy and correctness of the landmark extraction process.

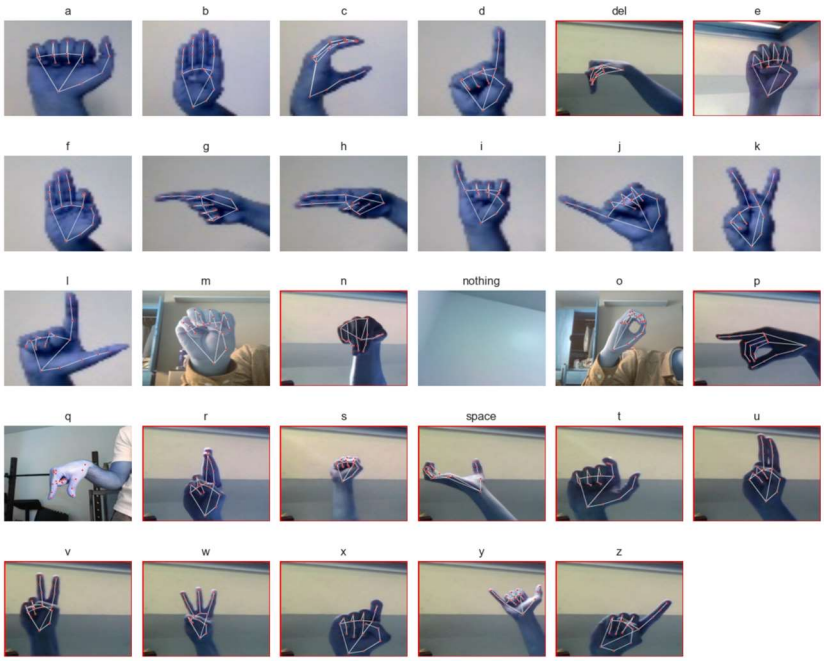

Figure 3 - In this visualization, we have selected an image from the dataset for each letter, including the 'space' and 'delete' signs. For each letter, the landmarks extracted from the image are aligned with the actual image. By overlaying the landmark.

3.b. Distribution of dataset

3.b.1. Distribution of labels

Distribution of labels:

f	21292	d	17215	e	15222	q	11886
l	21019	r	17177	w	15054	m	11585
k	20111	j	16324	x	14933	n	10700
g	19744	u	15860	s	14746	space	7568
i	19716	t	15764	c	14557	del	5308
h	19669	y	15566	z	13461	test	18
v	17648	b	15385	a	13233	nothing	8
		o	15346	p	12529		

Note: *We can see that there are two labels, namely "test" and "nothing," which correspond to random images within the dataset and are considered irrelevant for our analysis. Therefore, to ensure the accuracy and relevance of our data, we will drop these labels from the data frame.*

3.b.2. Preprocessing steps taken before continuing:

- Dropped the samples that have the labels “test” and “nothing”, as they’re irrelevant to our analysis here.
- Dropped duplicates as identified in the “image_name” column.
- Dropped “image_name” column, saved elsewhere.
  ⇨ *Final dimensions of the dataset: (109750, 65)*

3.b.3. Assess balance of the labels visually

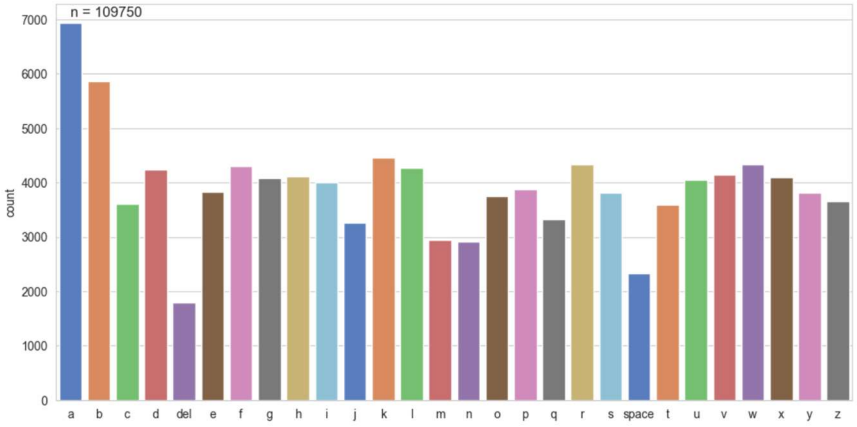

Figure 4 - We observed variations in the frequencies of images for different letters. The letter 'del' has the lowest frequency, with approximately 2,000 images, while the letters 'a' have the highest counts, with ~7,000 images. Although the 'del' class has the lowest number of images, we believe that it still provides enough data for training a reliable model.

3.b.4. Landmark Distribution and Depth

Since the origin of the z coordinate is set at the wrist, the landmarks closer to the camera will have smaller z values, while landmarks farther away will have larger z values. By analyzing the distribution of the z coordinates of the landmarks, we can gain a better understanding of the spatial positioning and variability of the images in our dataset. Therefore, we will observe the distribution of the z coordinates, to gain insights into how the landmarks vary across different images. Landmarks that are similar across images are expected to have a narrower distribution of z values, while landmarks that differ significantly will exhibit a wider distribution.

We will present boxplots showcasing the distribution of the z coordinate (depth) for each landmark per letter.

Figure 5 - As anticipated, we observe significant variability in most of the landmark's depth across the images of all the letters. This diversity in depth is expected to play a crucial role in learning the distinctive patterns associated with each letter, enabling the development of a robust and accurate model for sign language classification.

III. Baseline model – Random Forest Classifier

1. Model evaluation and implementation

1.a. Evaluation metrics

- **Accuracy**: Our goal is to have accurate results, classifying them correctly, not incorrectly.

1.b. Description of baseline model - Random Forest Classifier

Hyperparameters:

- *n_estimators=100*
- *n_jobs=-1*
- *[random_state=613]*

1.c. Input features

- 63 features:
  - 3 dimensions (x, y, and z) for each of 21 landmarks

1.d. Results:

METHOD	CORRESPONDENCE SUPERVISION	ACCURACY
RANDOM FOREST CLASSIFIER	**Matching pairs**	**0.95**

	precision	recall	f1-score	support
a	0.92	0.96	0.94	1423
b	0.95	0.98	0.97	1182
c	0.95	0.97	0.96	693
d	0.97	0.97	0.97	898
del	0.96	0.98	0.97	352
e	0.94	0.95	0.95	777
f	0.97	0.97	0.97	864
g	0.94	0.96	0.95	777
h	0.97	0.97	0.97	807
i	0.96	0.95	0.96	813
j	0.98	0.94	0.96	629
k	0.97	0.96	0.97	889
l	0.99	0.98	0.98	822
m	0.89	0.91	0.90	591
n	0.91	0.92	0.92	580
o	0.96	0.95	0.95	737
p	0.98	0.96	0.97	783
q	0.95	0.97	0.96	653
r	0.93	0.91	0.92	899
s	0.97	0.93	0.95	806
space	0.93	0.98	0.96	465
t	0.96	0.96	0.96	714
u	0.89	0.90	0.90	769
v	0.93	0.92	0.93	843
w	0.97	0.95	0.96	899
x	0.94	0.94	0.94	779
y	0.98	0.96	0.97	743
z	0.97	0.92	0.94	763
accuracy			0.95	21950
macro avg	0.95	0.95	0.95	21950
weighted avg	0.95	0.95	0.95	21950

Figure 6 - Based on the classification report, the model achieved an overall accuracy of 95%. Among all the classes, the letter 'l' had the highest precision of 0.99. On the other hand, the letters 'm' and 'u' had the lowest precision of 0.89 In terms of recall, the letters 'b', 'l' and 'space' had the highest value of 0.98. Conversely, the letter 'u' had the lowest recall of 0.90, indicating that the model missed a portion of the actual occurrences of 'u' in the data. The letter 'l' achieved the highest F1-score of 0.98, indicating a good balance between precision and recall for this class. Overall, the model performs well with high accuracy and balanced F1-scores across most classes. However, it exhibits slightly lower precision and recall for the letter 'u'.

IV. Next Steps

1. General steps:

1.a. Data Preprocessing to prevent overfitting

Part 1: Currently despite having a lot of images, the images do not differ enough, that's since the pictures are frames of videos. The result is our training and test set do not have enough variance and we end up with a very overfitted baseline model. The plan is to reduce the number of repeating pictures, diversify our training set, and get a test set that will better represent new data.

Part 2: Data augmentation, we're going to try oversampling our data that does not repeat as much and augment it slightly, that way our dataset should be able to tackle the overfitting problem we have.

1.b. CNN model to reclassify the landmarks

Assuming our assumptions about the data leakage and the lack of diversity, we want to create a 2nd model that will use CNN to be able to classify the landmarks after they have been remapped to a 3D representation.

1.c. EDA/Preprocessing

Our main goal for the project is to be able to translate hand gestures into letters that form words from a video that has been transformed to landmarks as seen in the Kaggle competition. In this part we'll dive deeper in the competition data to hopefully get a better understanding which model architecture to try.

2. Model selection:

2.a. Transformer models

It seems likely from the translation problem, and the impression we gather from the competition discussion page (as well as a similar competition's leaderboard) that we'll have to use some sort of Transformer model. Here are the model architectures that we are considering:

- BERT +Raw landmarks + timestamps.
- BERT + Raw landmarks + CNN model suggestion + timestamps.
- ViT Transformer + 3D landmarks + timestamps.
- ViT Transformer + 3D landmarks + CNN model suggestion + timestamps.
- LSTM + Raw landmarks + Random Forest suggestion.

Our goal will be to see what kind of model we'll be able to build that will give us the best chances.

3. Questions and reflections:

We suspect giving the transformer another input as a suggestion might help the model quite a lot assuming our suggestion makes sense and doesn't misclassify, the problem is that we're going to have new characters that we can't seem to find datasets that cover them. The question is how much will a suggestion from an incomplete model help in your opinion?

- Should we start trying to work on the Kaggle competition directly without going through the hassle of making the supplement CNN model as a proof of concept that we can predict letters from a hand pose/ as a supplement feature to help our final model.