📓 Week 1 Report — Data Collection, Preprocessing, and Exploration

Project Title:

Hand Gesture Recognition System

🧠 1. Objective

The primary objective for Week 1 was to **prepare a clean and structured dataset** that can be used for model training in later stages.
This involved collecting raw hand gesture data, cleaning it, removing noisy samples, and applying preprocessing techniques to make the data uniform and ready for deep learning pipelines.

📦 2. Tasks Completed

2.1 Data Collection

- Downloaded and organized a large dataset of Arabic sign language hand gestures.
- The dataset consisted of multiple images and labels stored in a structured format (images/, labels/, train.txt, and val.txt).
- Verified dataset integrity and ensured that all label files corresponded to existing images.

2.2 Data Cleaning & Background Removal

- Implemented background removal using the **GrabCut algorithm (OpenCV)**.
- The GrabCut method was chosen for its ability to accurately isolate hands even under variable lighting conditions.
- Result: Clean images with the hand region preserved and the background replaced with a uniform white background.

Tools used:

- Python, OpenCV, NumPy

Output Directory:

data/sign_data/cleaned_grabcut/

2.3 Image Filtering

- Some images were incomplete, blurry, or contained only parts of the hand or body.
- A filtering script (filter_bad_images.py) was implemented to automatically:
    - Detect and remove images with too much background or no visible hand.
    - Retain only high-quality images suitable for training.

Results:

Stage	**No. of Images**
Original	14,202
After Filtering	**11,670**
Removed	2,532 (≈18%)

Output Directory:

data/sign_data/filtered/

2.4 Dataset Splitting (Train / Validation)

- The cleaned dataset was divided into training and validation sets using an 80/20 ratio.
- Scripts used:
    - update_train_val.py
    - update_train_with_augmented.py

Resulting files:

- train.txt → 9,336 training images
- val.txt → 2,334 validation images

2.5 Data Augmentation

- Applied image augmentation to increase dataset diversity and prevent overfitting.
- Techniques applied:
    - Rotation (±20°)
    - Horizontal flipping
    - Zooming (±15%)
    - Brightness adjustment (±20%)
    - Translation (shift by 10%)
- Each image generated **3 augmented versions**, tripling the training data.

Result:

Dataset Type	**Image Count**
Original	11,670
After Augmentation	**≈44,341** total training images

Script used: augment_images.py

2.6 Final Dataset Organization

The dataset structure after preprocessing and augmentation was finalized as follows:

```
data/
 ├── sign_data/
 │   ├── filtered/      ← Cleaned images
 │   ├── augmented/     ← Augmented dataset
 │   ├── labels/        ← YOLO format labels
 │   ├── train.txt      ← Training image paths
 │   ├── val.txt        ← Validation image paths
 │   └── sign.yaml      ← Dataset configuration file
```

---

3. Tools and Libraries Used

Tool / Library	Purpose
Python 3.10+	Programming language
OpenCV	Background removal and image processing
TensorFlow / Keras	Image augmentation
NumPy	Numerical operations
os / glob	File handling and directory management
GitHub	Version control and collaboration

---

4. Results Summary

Process	Output
Background Removed	✅ Clean, uniform white background
Invalid Images Filtered	✅ 2,532 images removed
Dataset Split	✅ 9,336 (train), 2,334 (val)
Augmentation Completed	✅ Total 44,341 images
GitHub Repository Setup	✅ Project structure ready
Drive Dataset Upload	✅ Google Drive link shared for team use

Final Dataset Size: ~44K images (ready for model training)

🧩 5. Challenges and Solutions

Challenge	Solution
Background similarity between hand and environment	Used **GrabCut** instead of simple HSV segmentation
Dataset imbalance	Applied extensive augmentation to balance all gesture classes
Large data size for GitHub	Hosted dataset on **Google Drive** and linked in README

📘 6. Conclusion

Week 1 objectives were **fully achieved**.
A comprehensive and high-quality dataset was prepared for model development.
The data is now structured, cleaned, augmented, and split for training and validation.
All preprocessing scripts are functional, versioned, and documented for seamless collaboration in the upcoming weeks.