Sri Lanka Institute of Information Technology

**Final Report**

**2025-Y2-S1-MLB-B1G2-07**

| | |
|---|---|
| IT24104110 | Thulmanthi W.A.S |
| IT24104081 | Gurusingha R.N |
| IT24104133 | Vaaranan J |
| IT24104068 | Chandrasiri R.M.D.S |
| IT23379138 | Rajamuni R.D.V.R |
| IT24610827 | Lahiruni K.L.M |

**Artificial Intelligent and Machine Learning| IT2011**

B.Sc. (Hons) in Information Technology

# Introduction and Problem Statement

**Motivation**
Over 466 million people worldwide live with disabling hearing loss (WHO, 2021). For many, sign language is their primary mode of communication. However, non-signers often struggle to interact with the hearing-impaired community, leading to social, educational, and professional exclusion.

**Problem Statement**
Communication between hearing-impaired individuals and non-signers remains slow, inefficient, and heavily reliant on human interpreters—resources that are often unavailable in real-time settings like hospitals, schools, or public transport.

**Proposed Solution**
We developed a real-time sign language digit recognition system using a Convolutional Neural Network (CNN) that classifies static hand gestures representing digits 0–9 from live video input. This system serves as a foundational step toward broader sign language interpretation tools powered by AI.

**Project Goal**
To design, train, and validate a Convolutional Neural Network (CNN) capable of accurately classifying static hand gestures representing digits 0–9.
The goal is to achieve ≥ 90 % test accuracy with robust performance under varying lighting and orientation conditions, and to deploy the model for real-time inference through webcam-based video streams.

# Dataset Description

- Source : Kaggle 's Sign language Digits Dataset https://www.kaggle.com/datasets/ardamavi/sign-language-digits-dataset
- Structure : 10 folders (0-9),each containing grayscale images of hand signs of relevant digit.
- Size : 200-250 images per class (total ~2,200 images)
- Image Format : 100 x 100 pixels with color space RGB
- Target : Digit label (0-9)

🔗 **Dataset Preview:**

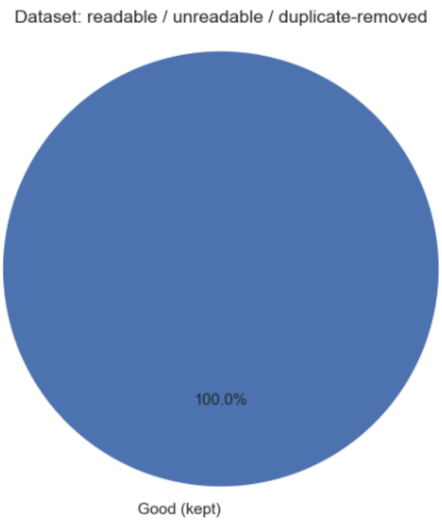

**Details of datasets:**

- Image size: 100 x 100 pixels
- Color space: RGB
- Number of classes: 10 (Digits: 0-9)
- Number of participant students: 218
- Number of samples per student: 10

# Preprocessing & EDA

### Data Validation and Cleaning

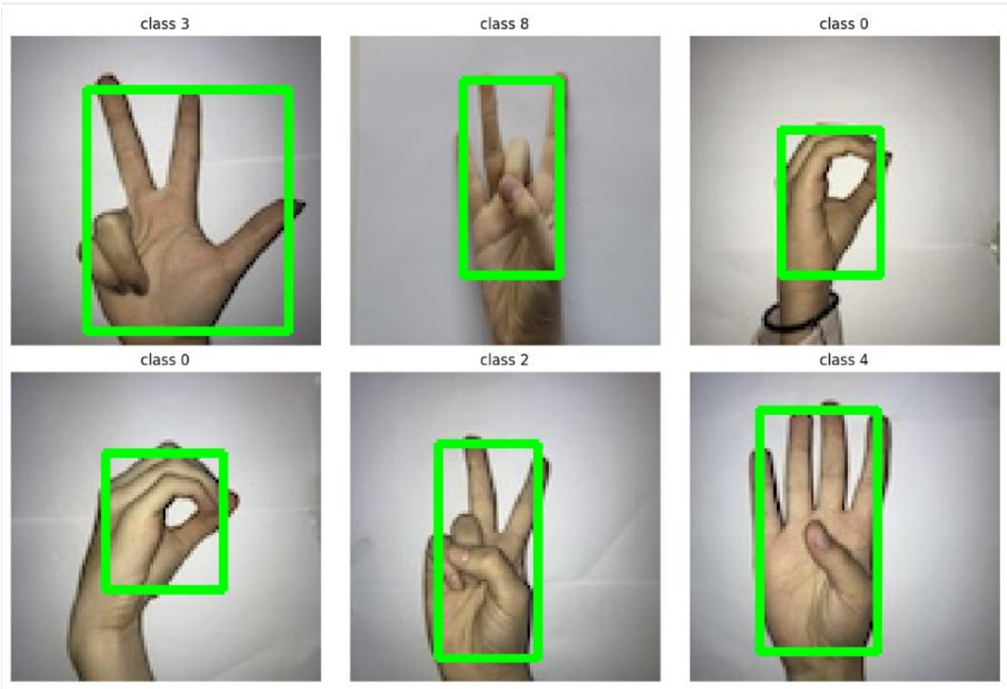
- File Verification: Checked the validity of all image files to identify corrupted, unreadable, or duplicated entries.
- Dataset Quality Check: Ensured no broken or repeated images existed in the dataset to maintain data integrity.

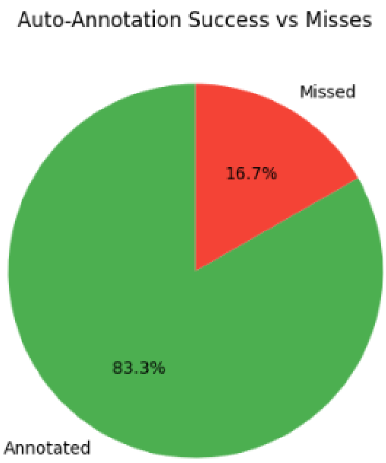


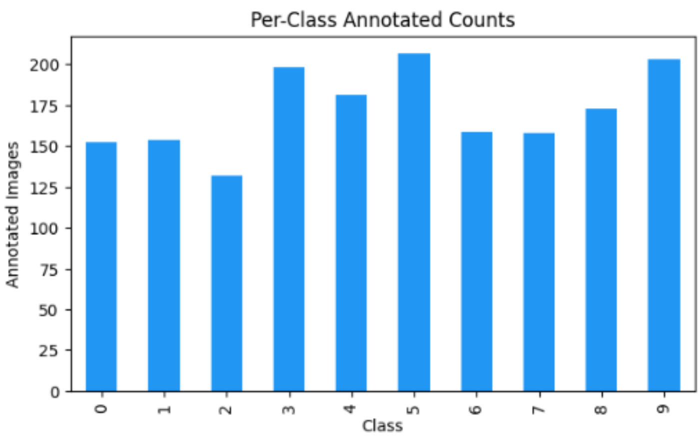

```
SUMMARY:
  Raw discovered files: 43
  Readable (kept before dedupe): 2062
  Unreadable (bad): 0
  Deduped: 2062 -> 2062 (removed 0 duplicates)
  Final manifest entries: 2062
  Per-class counts (after dedupe):
cls
0     205
1     206
2     206
3     206
4     207
5     207
6     207
7     206
8     208
9     204
```

**Image Annotation**

- A pre-trained model was utilized to automatically annotate images by detecting hand bounding boxes. Manual annotation for all 2062 images was not feasible; therefore this automated approach accelerated the labeling process.
- Images successfully annotated were retained, while missed or unannotated images were dropped to maintain consistency.



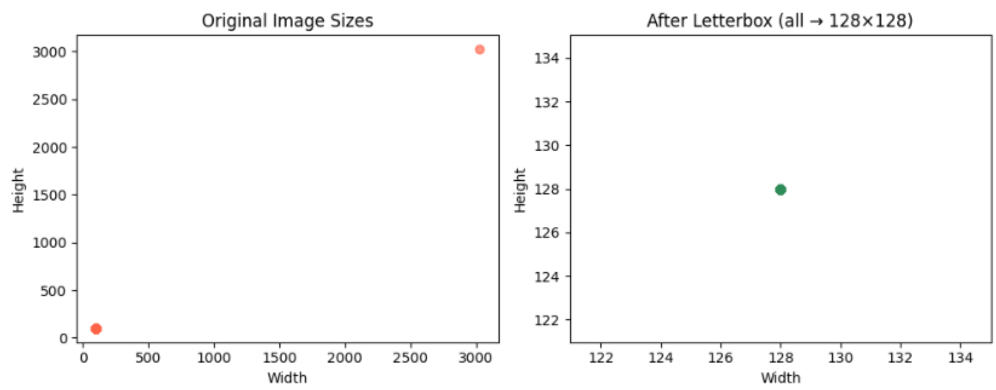




**Square Cropping and Resizing**

- The detected bounding boxes were cropped and resized to 128×128 pixels.

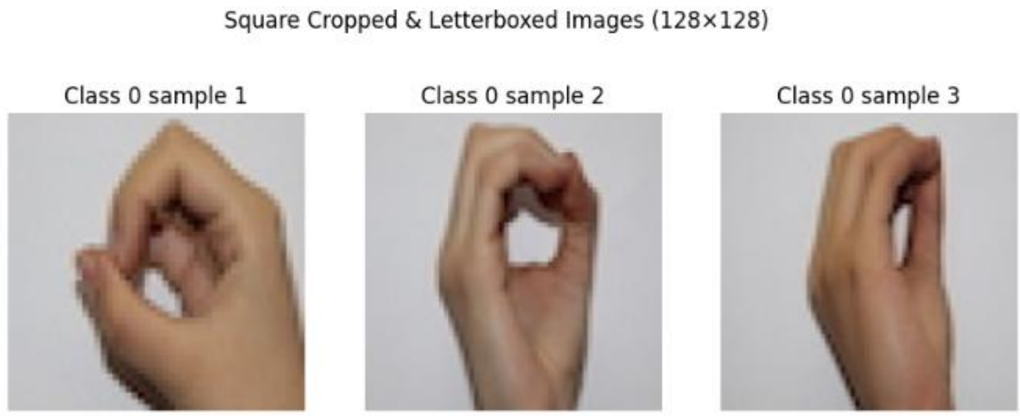
- This standardization was necessary since deep learning models require fixed-size inputs for effective training.
- The resizing ensured that all images were uniform, allowing the neural network to learn consistently across samples.




### Image Quality Filtering

To enhance dataset robustness, only clear and high-quality hand sign images were retained:

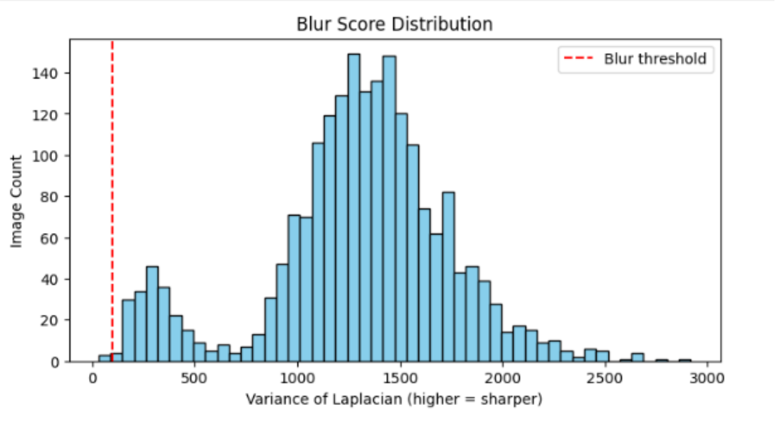
- Blur Detection: Applied Variance of Laplacian — lower values indicated higher blur, and such images were removed.

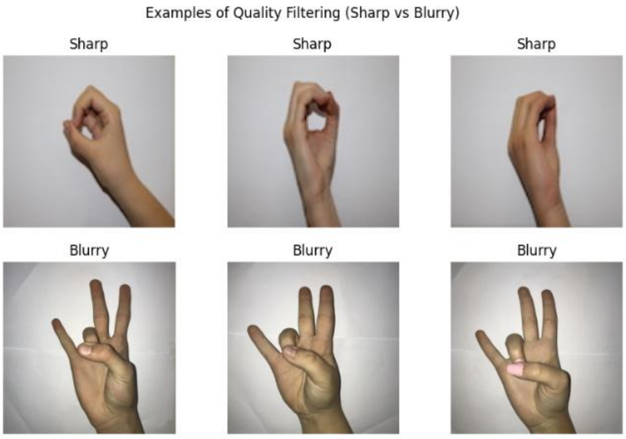


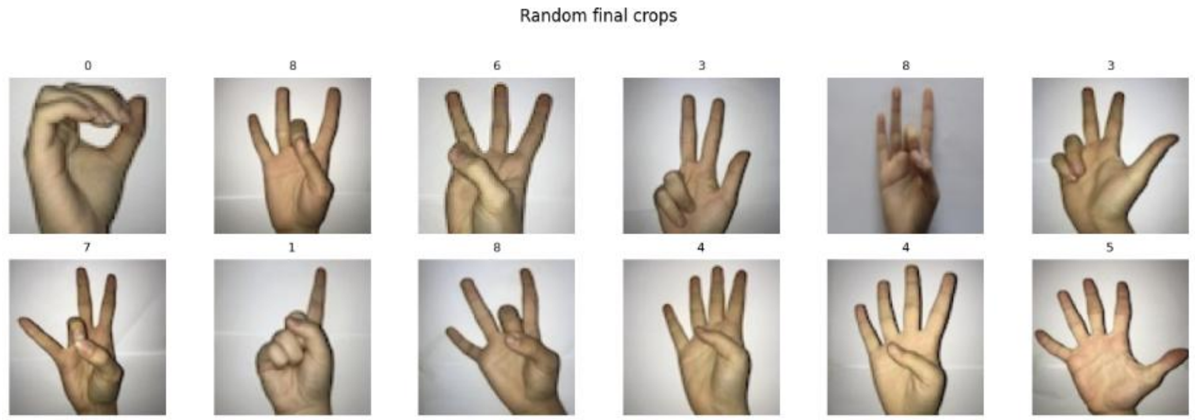

- Darkness Detection: Calculated mean grayscale intensity-images below a defined brightness threshold were discarded.
  These steps ensured that blurry or dark images, which can confuse the model and reduce feature extraction accuracy, were eliminated.

**Outlier Removal**

- This step focuses on identifying and eliminating anomalous data points from the overall data distribution. Outliers can distort statistical summaries and negatively impact model accuracy. In this process, statistical techniques such as the Interquartile Range (IQR) method are applied to detect values that fall outside the acceptable range for each feature.



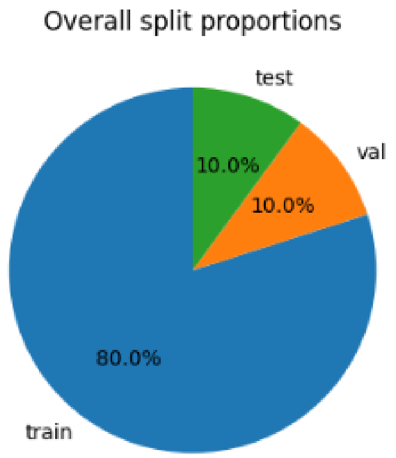



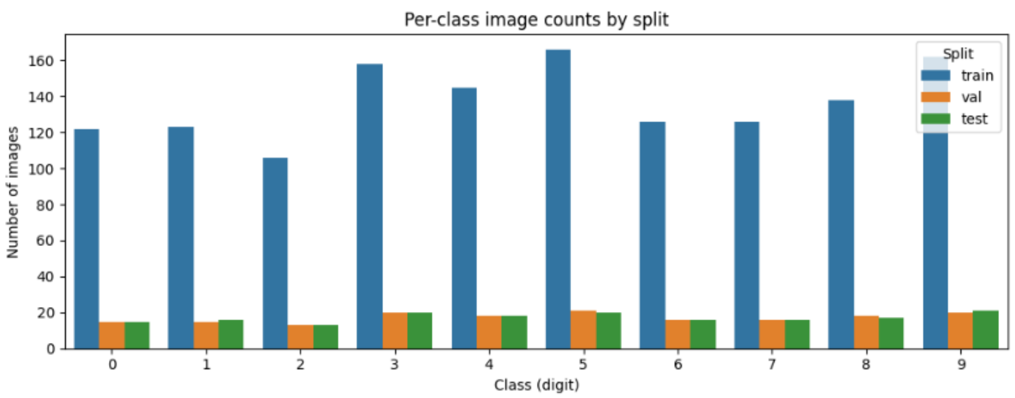




**Dataset Splitting**

The cleaned dataset was split into training, validation, and testing subsets using stratified sampling:

- Split Ratios: 80% Training, 10% Validation, 10% Testing.
- Stratification: Guaranteed each digit class (0–9) maintained equal representation across all splits.
- Reproducibility: Ensured through a fixed random seed (random state = RNG_SEED).

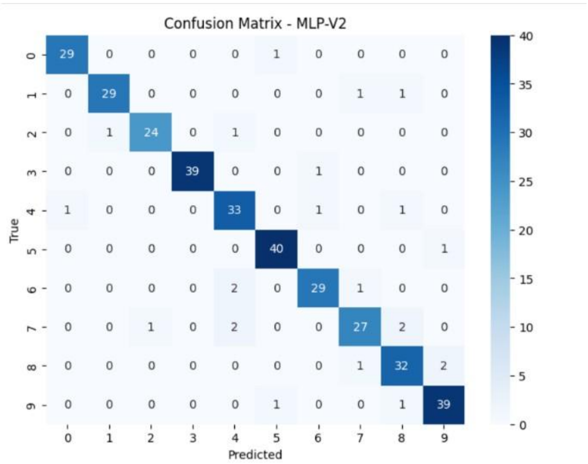


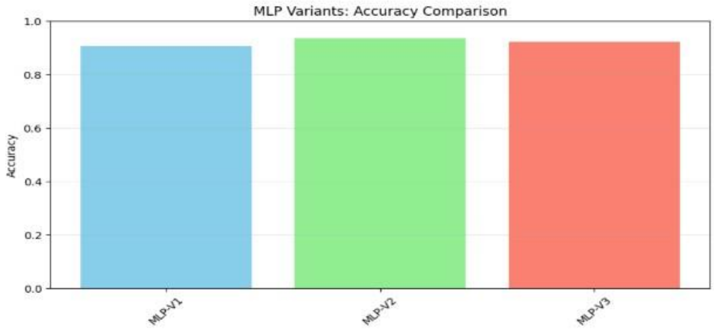



# Model Design and Implementation

1. **Multi-Layer Perceptron (MLP)** **:** IT24104068 - Chandrasiri R.M.D.S
   - ➢ Description

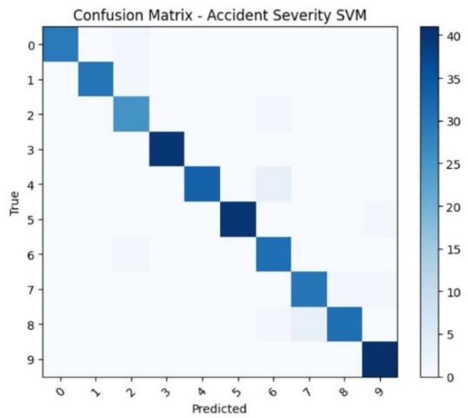
     MLP is a type of feedforward neural network with one or more hidden layers between the input and output. Each neuron is connected to all neurons in the next layer, and it uses non-linear activation functions (like ReLU or Tanh) to learn complex patterns.
   - ➢ Hyperparameters
     - Hidden layers: (128, 64)
     - Activation: Tanh
     - Solver: Adam
     - Max iterations: 200




2. **Support Vector Machine (SVM)** : IT24104133 - Vaaranan J
   - ➢ Description

     Support Vector Machines (SVMs) are well-suited for training on our dataset because they handle high-dimensional data effectively—common when images are flattened into feature vectors—and can model complex, non-linear class boundaries using kernel functions. SVMs also generalize well with limited data and rely only on support vectors after training, making them memory-efficient.
   - ➢ Hyperparameters
     - Kernel : RBF
     - Class Weighting: set to ‘Balanced’
     - Random State: fixed to 42
     - Kernel Coefficient : default value used gamma=’scale’



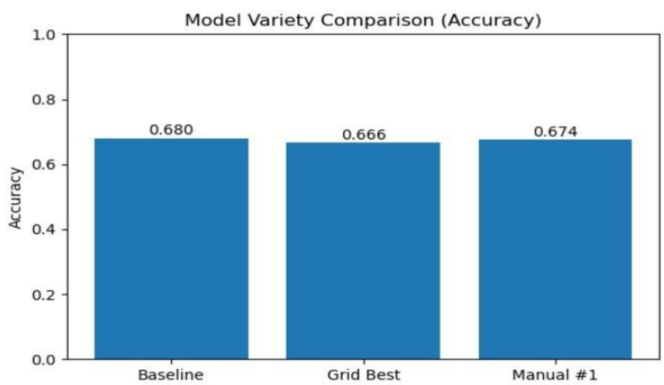

3. **Decision Trees : IT23379138 - Rajamuni R.D.V.R**

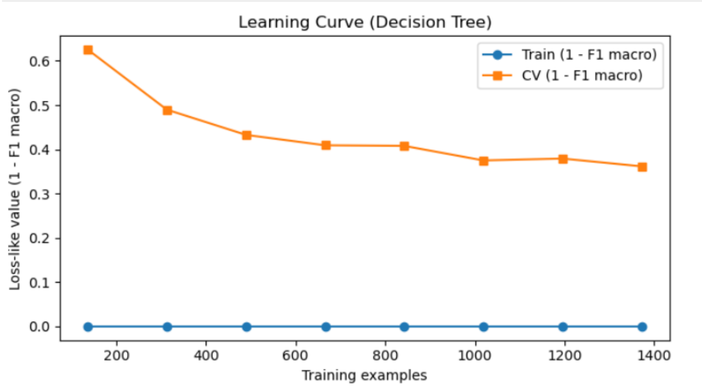
➢ Description

A decision tree is a non-parametric supervised learning algorithm that splits data based on feature thresholds to classify samples. It's easy to interpret and handles both numerical and categorical data well. We trained 3 variants: a baseline, a manually tuned version, and the best model from GridSearchCV

➢ Hyperparameters

- Criterion = 'gini'
- Max_depth = NaN
  Min_sample_leaf = 1
- ccp_alpha : 0.0000




4. **CNN : IT24104110 – Thulmanthi W.A.S**

➢ Description

A Convolutional Neural Network (CNN) is a type of deep learning model specially designed for processing grid-like data, such as images. It automatically learns spatial hierarchies of features — from edges and corners in early layers to complex patterns (like hands or digits) in deeper layers.

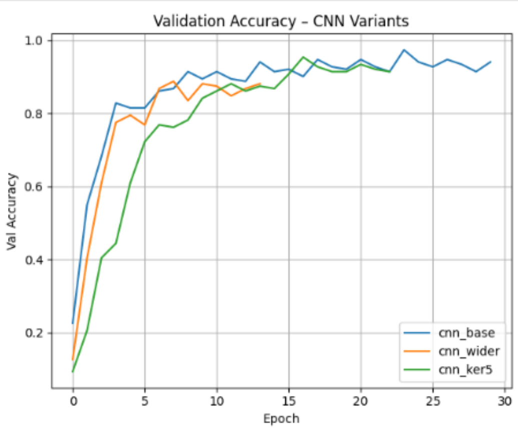
➢ Hyperparameters

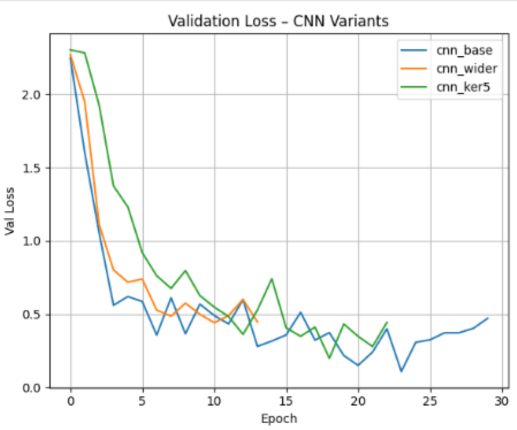
- Architecture – filters: (32, 64, 128) / (64, 128, 256)
  Kernel Size: 3x3,5x5

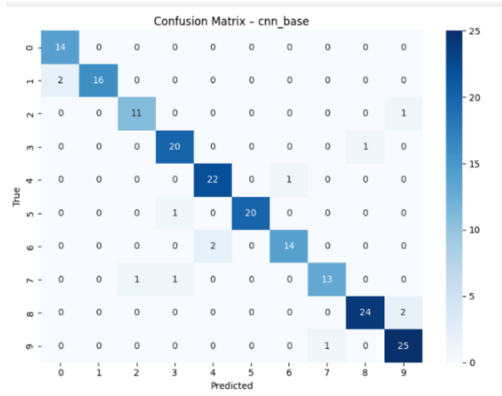
Dropout : 0.25,0.30

- Training –  Batch Size: 64

  Epochs : 30
- Regularization – EarlyStopping,ReduceLROnplateau
- Validation – K-Fold : 3 folds on training dataset



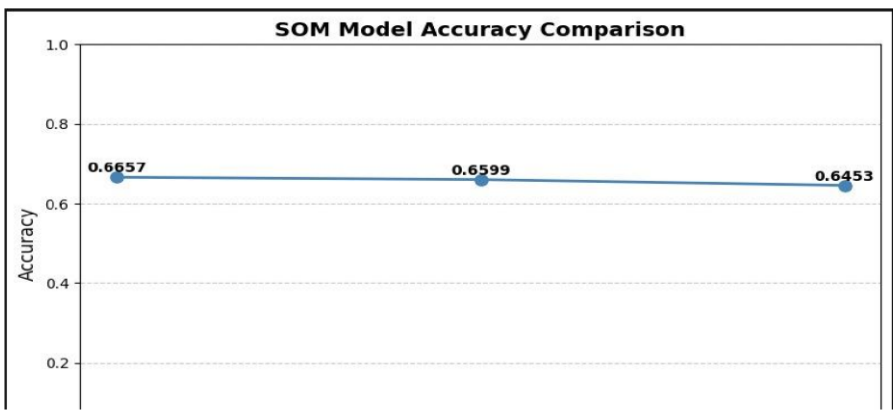




## 5. SOM : IT24104081 – Gurusingha R.N

- Description

  A Self-Organizing Map (SOM) is an unsupervised neural network that reduces high-dimensional data (like images) into a simple 2D map while keeping similar data points close together. It's useful for visualizing patterns and grouping similar inputs without needing labeled data.
- Hyperparameters
  - Grid size – 8*8
  - Sigma – 1.0
  - Learning Rate -0.5
  - Validation – K-Fold cross validation



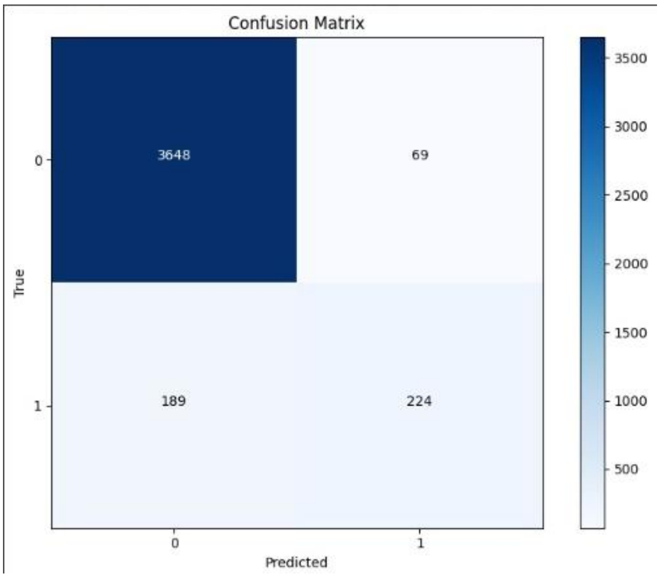
6. **KNN : IT24610827 - Lahiruni K.L.M**

➢ Description

KNN is a simple, instance-based algorithm that classifies a new sample by looking at the k most similar (nearest) samples in the training data. It then assigns the majority class among those neighbors as the prediction.

➢ Hyperparameters

- **Number of Neighbors (n_neighbors)**: [3, 5, 7, 9]
- **Weight Function (weights)**: [‘uniform’, ‘distance’]
- **Distance Metric (metric)**: [‘euclidean’, ‘manhattan’]



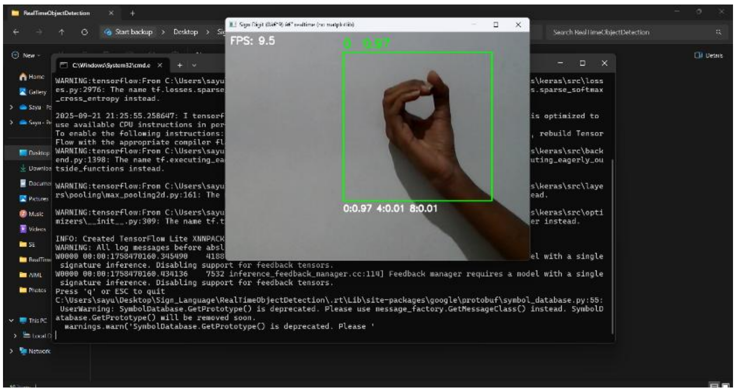

# Evaluation and Comparison

| **Model** | **Accuracy** | **Precision** | **Recall** | **F1 Score** |
|---|---|---|---|---|
| CNN | 0.9322 | 0.9332 | 0.9293 | 0.9300 |
| MLP | 0.9331 | 0.866 | 0.854 | 0.9322 |
| SVM | 0.9593 | 0.96 | 0.96 | 0.96 |
| KNN | 0.5423 | 0.773 | 0.543 | 0.628 |
| Decision Tree | 0.68099 | 0.69902 | 0.67430 | 0.67981 |
| SOM | 0.665 | 0.643 | 0.664 | 0.655 |

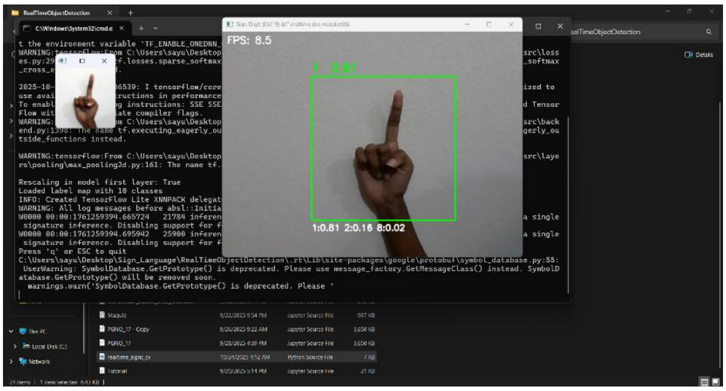
After evaluating multiple machine learning models,we selected the Convolutional Neural Network (CNN) as our final model due to its superior performance in both accuracy and precision. The CNN demonstrated robust classification capabilities on our hand sign recognition task, consistently outperforming other architectures. To validate real-time applicability, we integrated the trained CNN with OpenCV for live camera inference. The system successfully detects and classifies hand signs in real time, displaying bounding boxes with a accuracy percentage on it (“letterbox”-style detection) around recognized gestures.

**Sign language Digit Real Time Detection using CNN model**

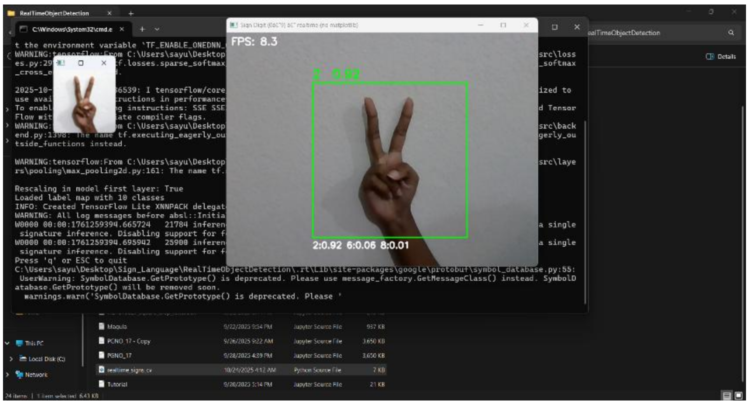
**Digit 0**



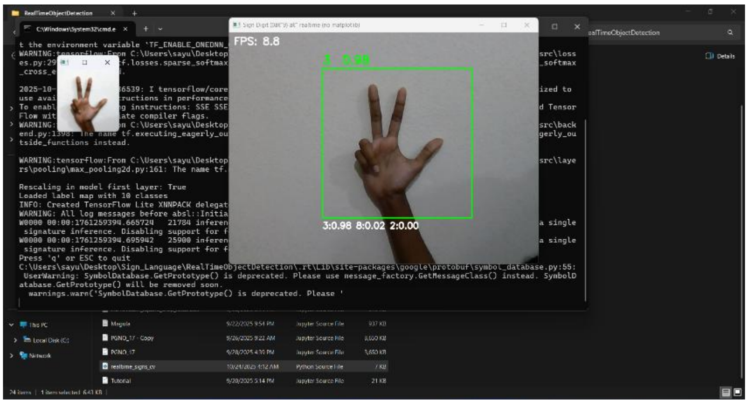

**Digit 1**



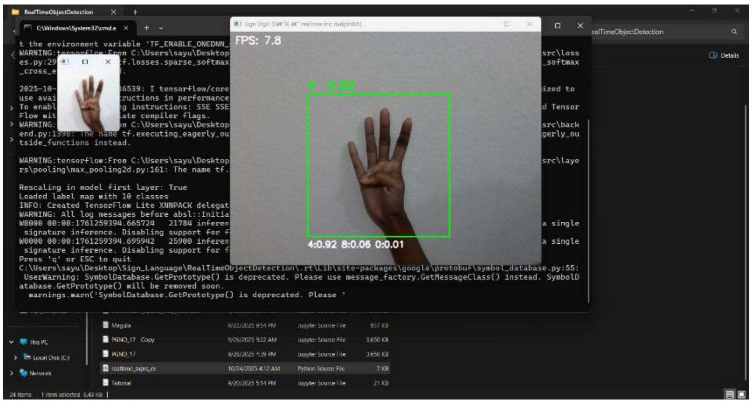

**Digit 2**



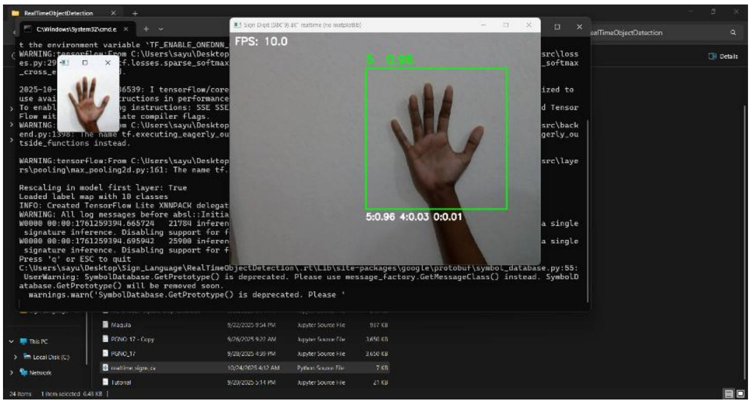

**Digit 3**



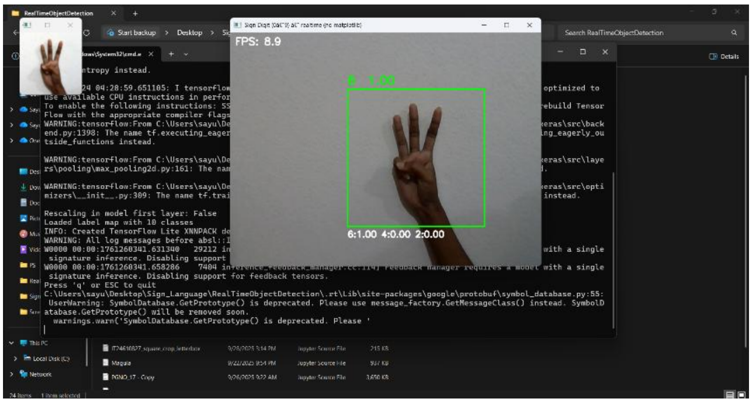
**Digit 4**



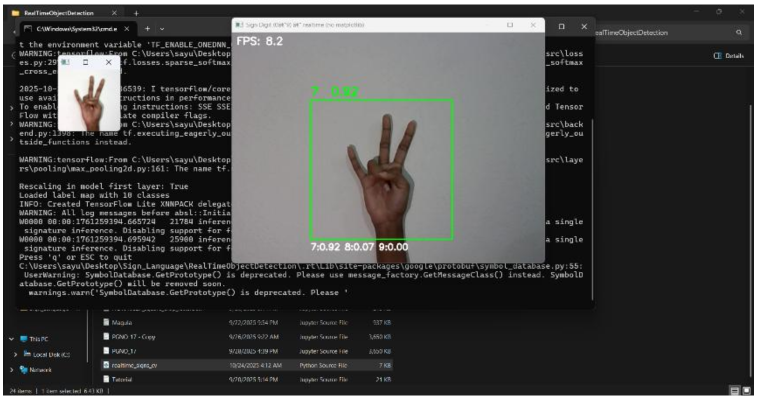

**Digit 5**



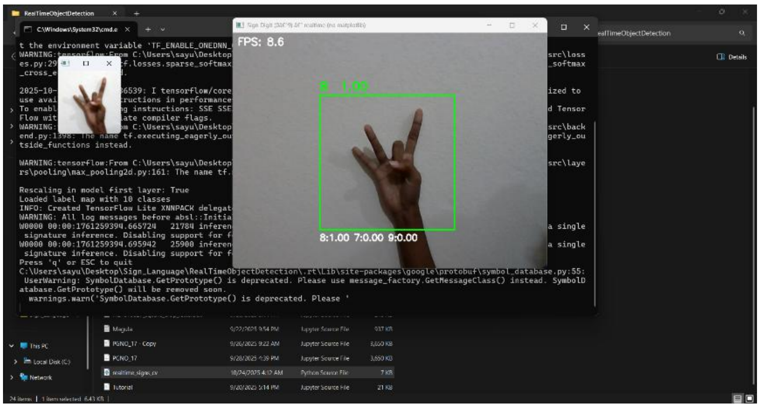

**Digit 6**



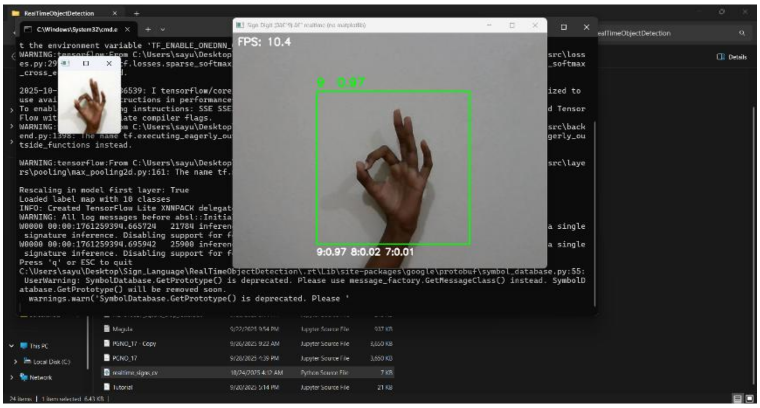

**Digit 7**




**Digit 8**




**Digit 9**



## Ethical Considerations and Bias Mitigation

Developing a Real-Time Sign Language Recognition system involves critical ethical responsibilities related to fairness, inclusivity, and responsible AI development.

**Bias and Fairness**:

Bias can arise from datasets that lack diversity in signers' skin tones, hand shapes, or lighting environments. To ensure fairness:

- Balanced Sampling: Include signers from varied backgrounds.
- Continuous Evaluation: Regularly test across different demographic groups.
- Data Augmentation: Simulate real-world diversity to improve model generalization.

**Ethical Implications:**

AI systems for sign recognition must be used ethically:

- Data Privacy: Obtain informed consent and anonymize all collected data.
- Transparency: Explain model predictions clearly to users.
- Accountability: Limit system use to educational or assistive purposes.

**Mitigation Strategies:**

- Maintain a diverse dataset.
- Use Explainable AI methods for transparency.
- Conduct regular bias audits and update datasets periodically.
- Involve the deaf community in testing and feedback.

# Reflections and Lessons Learned

This project provided valuable experience in the end-to-end process of building an AI-based recognition system.

**Technical Learnings:**

Learned how to preprocess image data effectively.

Understood the working of Self-Organizing Maps and supervised learning.

Improved knowledge of Python libraries such as NumPy, Matplotlib, tenserflow and OpenCV.

**Challenges Faced:**

- Handling images dataset.
- Achieving real-time processing speed.
- Challenge: Domain Gap Between Training Data and Real-World Conditions
  Our training dataset consisted of clean, cropped hand images with uniform white backgrounds and minimal visual variance. While this improved initial model accuracy under controlled conditions, it created a significant domain gap during real-time deployment. In real-world scenarios, hand signs appear against diverse, dynamic backgrounds with varying lighting, angles, and partial occlusions. Because the model was not exposed to such contextual complexity during training, its performance degraded when tested via webcam feed unless the background was manually controlled .To mitigate this, we had to carefully test the training conditions during live inference—limiting practical usability. This highlighted the critical need for training data that includes realistic backgrounds and environmental variability to ensure robustness in real-time applications.

**Future Improvements:**

To enhance real-world applicability, the dataset should be expanded to include greater visual diversity such as varied backgrounds, lighting conditions, hand sizes, skin tones, and signing angles to bridge the gap between controlled training environments and dynamic real-time settings. This will improve the model's robustness and generalization in uncontrolled scenarios. Additionally, the current system recognizes isolated signs in real time; a key next step is to implement sequential gesture recognition, enabling the system to capture and interpret continuous input such as digit-by-digit entry for phone numbers or short words. This would significantly increase the system's utility in practical communication tasks.