Dataglove

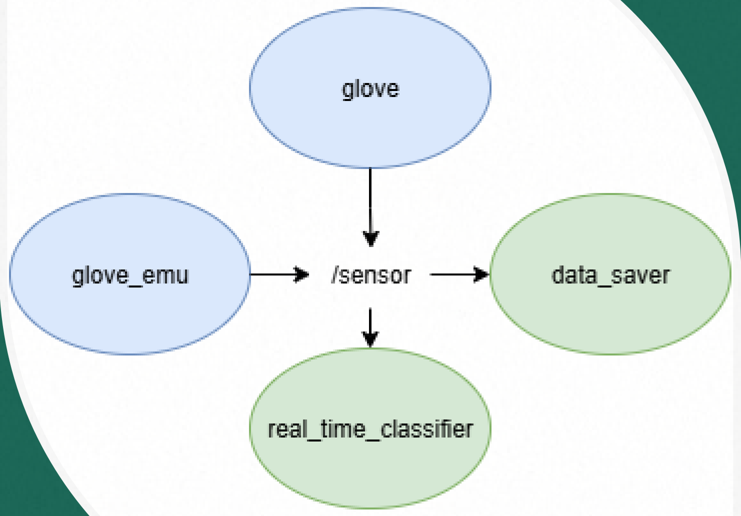
Real time object classification by grasping via sensorized glove

VMG30

- **Finger flex sensors (10)**
  Measure the bending angle of each phalanx
- **Abduction sensors (4)**
  Measure the angular distance between adjacent fingers
- **Pressure Sensors (5)**
  Detect the force applied with the fingertips
- **Structural Sensors (2)**
  Measure the overall posture of the hand
- **IMU Units (2)**
  Provide 3D spatial orientation

ROS2 architecture

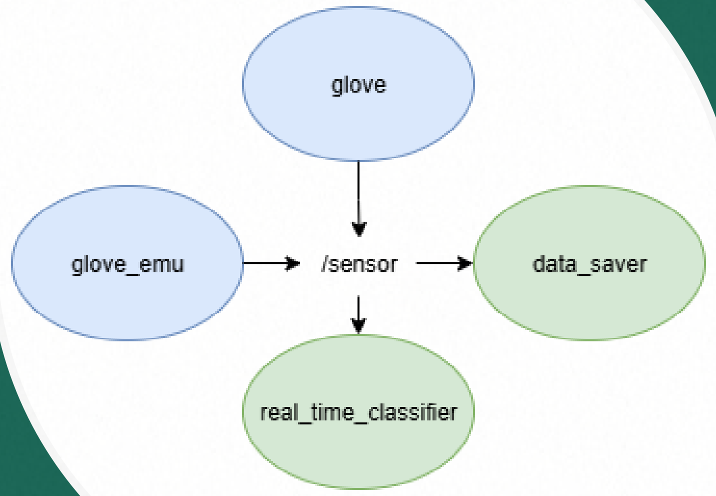
- Glove node
  - Launches **two parallel threads**:
    - **Thread 1**: reads data from the serial port
    - **Thread 2**: publishes data on the /sensor topic
  - Operating frequency: **100 Hz** for both threads
- Real time classifier node
  - Subscribes to the /sensor
  - Maintains a **rolling window of 300 time samples**
  - Uses the buffered data to **classify the object held** by the glove in real time



ROS2 architecture

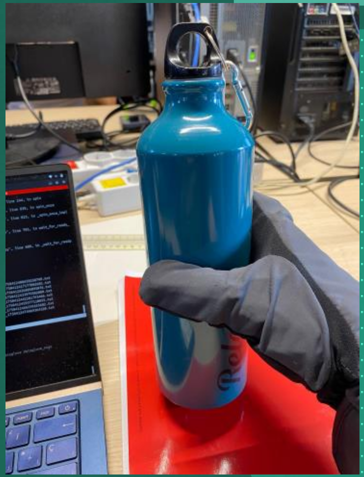
- Data saver node
  - Subscribes to the /sensor
  - Store to a file the received data
  - Used for creating the **dataset**

- Glove emu node
  - Read data from a file
  - Publish data to /sensor with a frequency of 100Hz
  - Used to **emulate** the glove



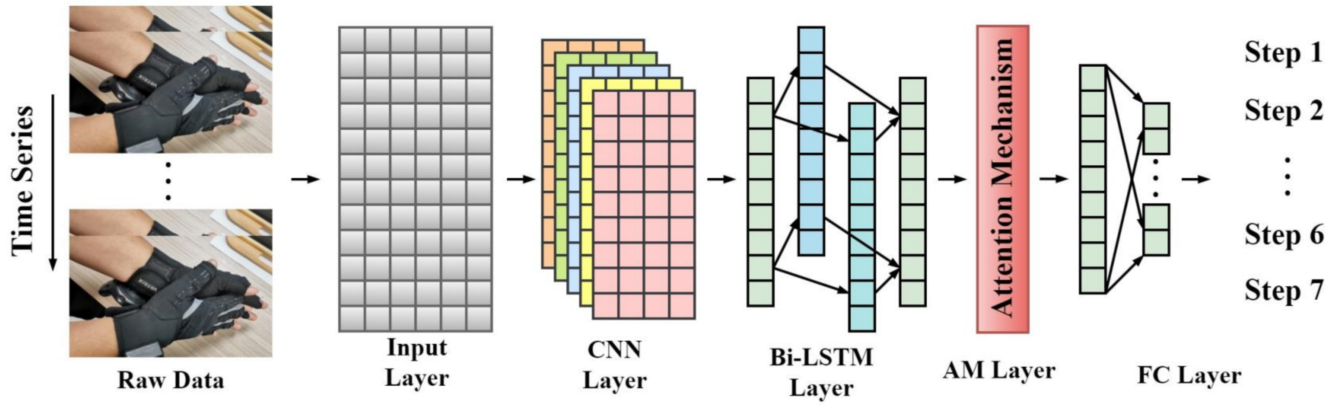
Dataset

- 6 object
  - No object (rest position)
  - Bottle
  - Pen
  - Phone
  - Computer mouse
  - Glasses case
- Execute glove node and data saver node:
  - ~ 1 second in rest position
  - Move and grasp object
  - ~ 1 second stationary with object in hand

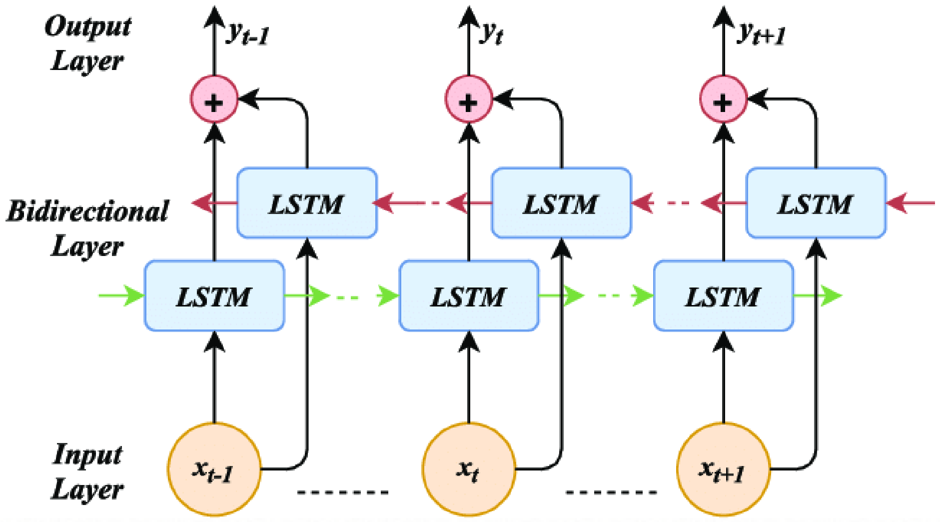

CNN-BiLSTM model with attention mechanism

- 2 convolutional layer
  - Kernel size 1x3
- 2 Bi-LSTM layer
- 1 Attention Layer
- 2 Fully connected layer



Bidirectional LSTM

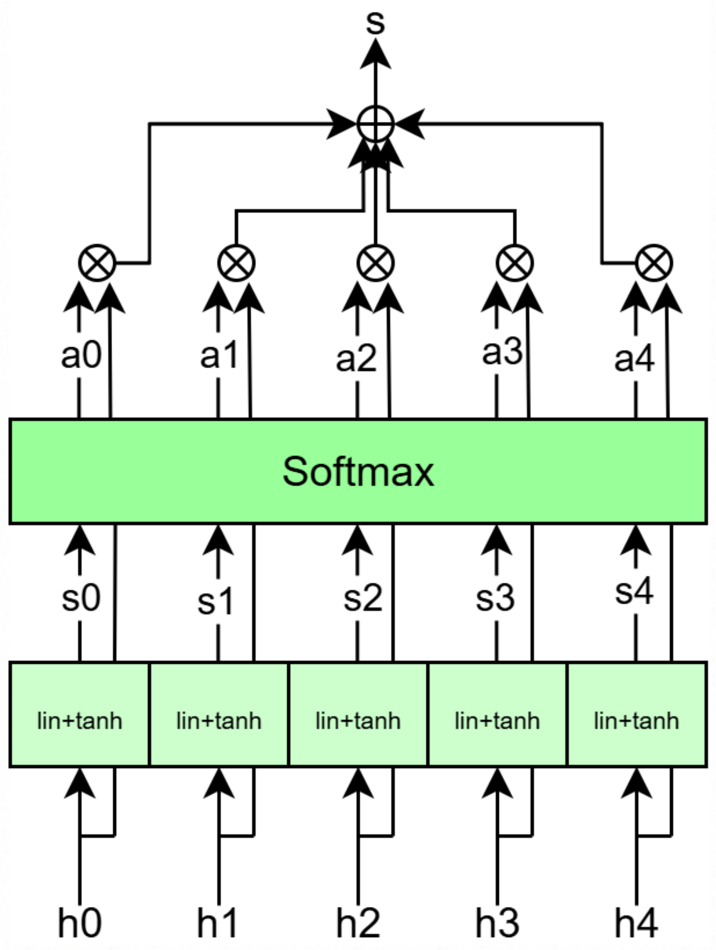
- LSTM are recurrent NN
- They use gating mechanism to mitigate gradient vanishing problem
- Bi-LSTM process input sequences in both forward and backward directions
- This allows the model to capture context from both past and future states



Attention mechanism

- Score computation:
  - $s_t = \tau(w_h * h_t + b_h)$
- Attention weights:
  - $a_t = \frac{\exp(s_t)}{\sum \exp(s_i)}$
- Context vector:
  - $s = \sum_t \ a_t h_t$

A weighted sum of the scores produces the final context vector, which focuses on the most relevant parts of the input sequence.



Model training

Dataset split:

60% – Training set

20% – Validation set

20% – Test set

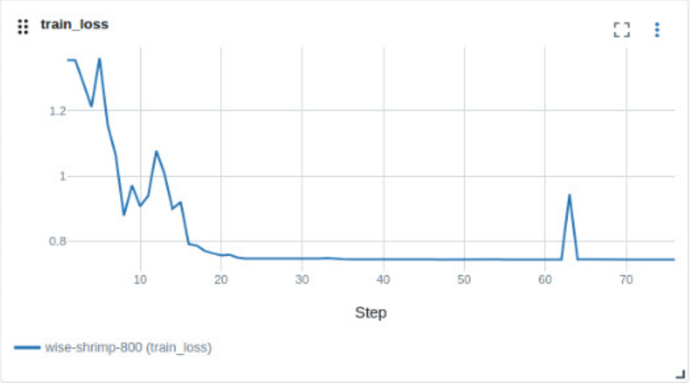
Validation set used for:

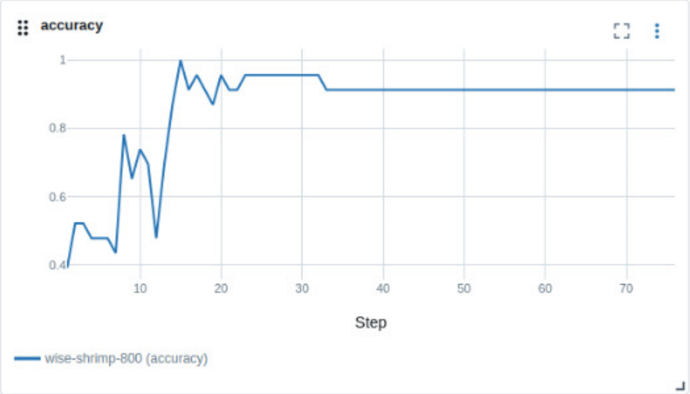
Monitoring model performance during training

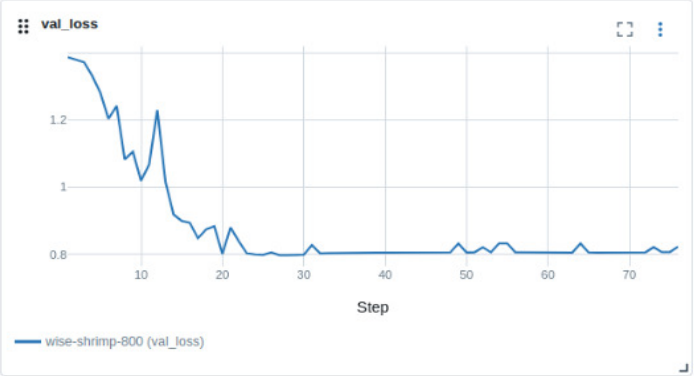
Implementing early stopping to avoid overfitting

Early stopping strategy:

Training is interrupted if no improvement in validation performance is seen for 50 consecutive epochs

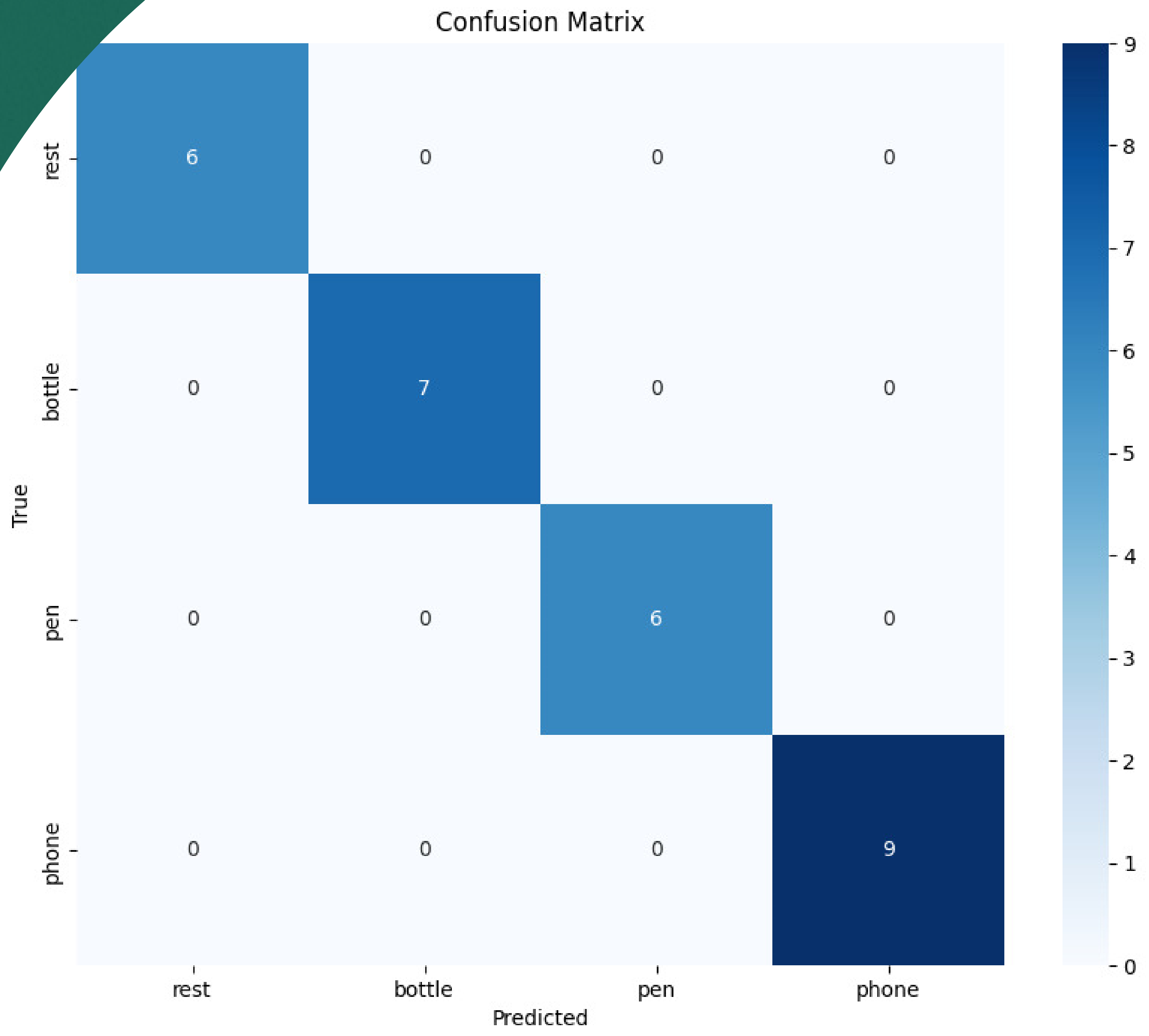




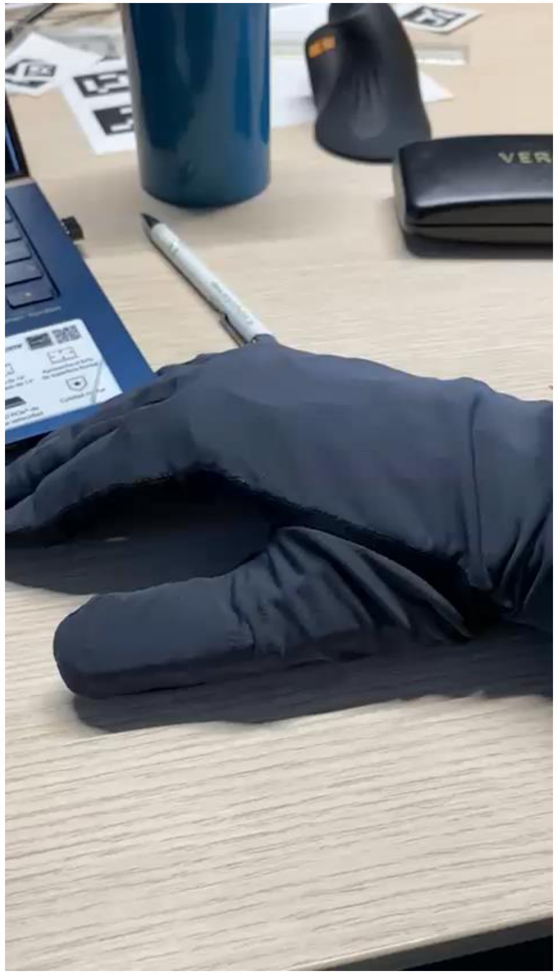


On test set

100% accuracy



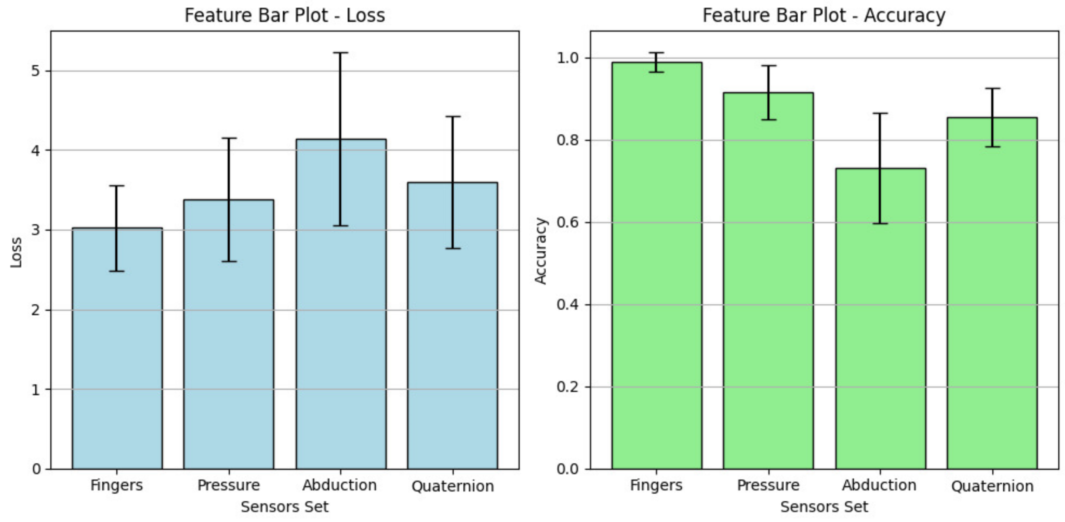
Demo Time



Feature curve and most relevant sensor type

- 5-fold cross validation
  - Points are the average value on validation sets
  - Error bar are standard deviation on validation sets

- **Finger flex data** are the most descriptive



Continual learning

Class incremental learning: I want my classifier to be able to learn new classes (last two object)

Avoid retraining from scratch

Faster

Cheaper

Attention: Catastrophic forgetting

Learning without forgetting

- Save a copy of the old model – **Teacher**
- Modify the FC layer of the model to add new classes and copy the shared weights – **Student**
- Train **only** on **new data**:
  - $L = L_{class} + \lambda L_{distill}$
  - **Task loss** for the new classes - CE
  - **Distillation loss** to retain knowledge of old classes - KL
- Doesn’t require Buffer Replay

LwF results

- If λ is too large, the model focuses too much on preserving old knowledge

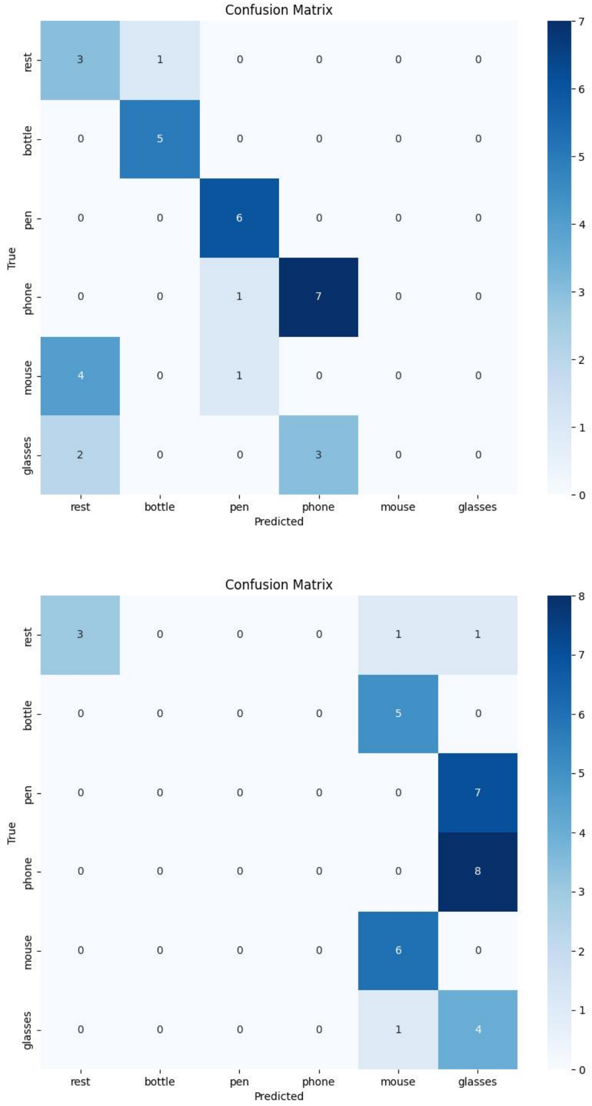
→ It fails to learn the new task properly

- If λ is too small, the model prioritizes learning new data

→ It forgets the previous tasks

→ Catastrophic Forgetting

- **Choosing the right value of λ** is essential to balance **stability** and **plasticity** but is hard



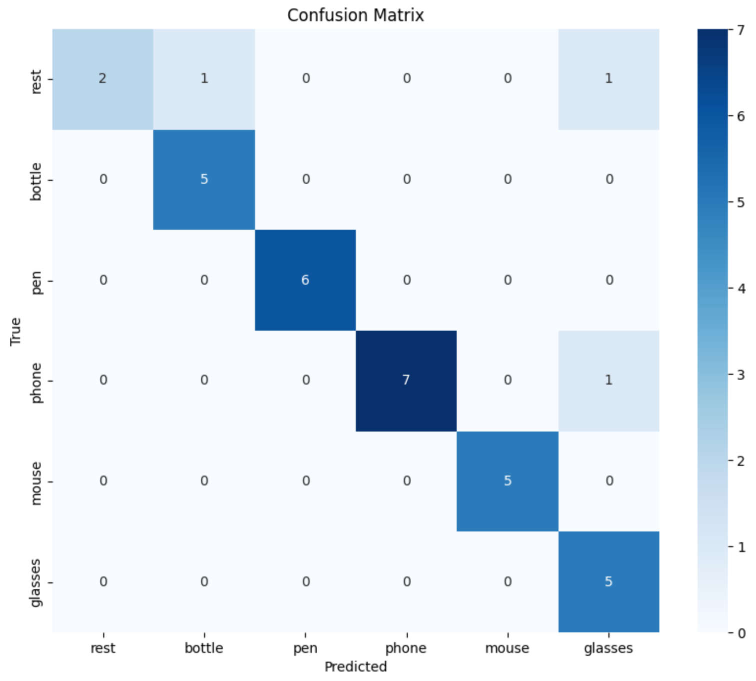
Dark Experience Replay

- Save a copy of the old trained model – **Teacher**
- Modify the FC layer of the model to add new classes and copy the shared weights – **Student**

- **Accessing a small fraction of old data – Buffer Replay**
- Train at the same time on new and old data but using different loss
    - $L = CE(f(x), y) + \lambda\, MSE(f(x_{old}), f_{old}(x_{old}))$
    - Classification term + distillation term

DER results

- 90% accuracy
- New classes are learned successfully
- **Buffer replay** is required but helps mitigate catastrophic forgetting
- Easier to choose λ



Conclusion

Implemented a state of the art neural network for classifying object via a sensorized glove

Implemented a ros2 program for real time classification

Compared two technique for continual learning:

- Learning without forgetting
- Dark experience replay

Thanks for you attention.

Bibliography:

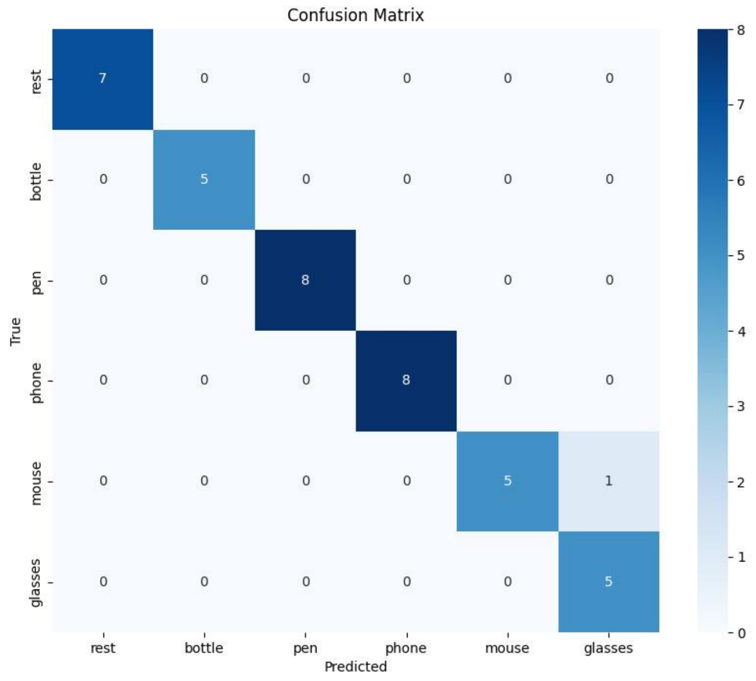
Data glove-based gesture recognition using CNN-BiLSTM model with attention mechanism, Wu Jiawei et al. (journals.plos)

Learning without Forgetting, Zhizhong Li et al. (arXive)

Dark Experience for General Continual Learning: a Strong, Simple Baseline, Pietro Buzzega et al. (arXive)

Without continual learning

- 97% accuracy
- Class incremental learning worsen the overall performance of the model



Feature curve and most relevant sensor type

- 5-fold cross validation
  - Points are the average value on validation sets
  - Error bar are standard deviation on validation sets

- **Finger flex data** are the most descriptive