INTERNATIONAL TELECOMMUNICATION UNION

TELECOMMUNICATION STANDARDIZATION SECTOR

STUDY PERIOD 2022-2024

FOCUS GROUP ON AUTONOMOUS NETWORKS (FG-AN)

AN-I-xx

Question(s):	NA	February 2023 (TBC)

INPUT DOCUMENT

Source:	TCS Smart Machine
Title:	Report on activities for Build-a-thon 2022 from TCS Smart Machine

Contact:	Chandan Singh	E-mail: chandan1214666@gmail.com
Contact:	Vipul Sanap	E-mail: vipulasanap@gmail.com
Contact:	Ayush Kumar	E-mail: ayushkumar94314@gmail.com

Keywords: 5G, Artificial Intelligence, build-a-thon, Challenge, closed loop, controller, hackathon, Machine Learning, Proof of concept

Abstract: This contribution provides a report on activities by the TCS SMART Machine team towards the Build-a-thon 2022. We analyze IIT-D use cases, “Slip Detection (and Force Estimation)” and “Object Detection” in a robotic arm, and produce a design as per the reference design in the Build-a-thon repository. We also provide the corresponding code based on the reference code in the Build-a-thon 2022 repository. After analysing the use cases, we trained two ML models, one for “Slip Detection (and Force Estimation)” and another one for “Object Detection” . We have tested and validated the models for the use cases.

Summary

This document provides a report on the analysis of IIT/D use cases, Slip Detection (and Force Estimation)” and “Object Detection” in a robotic arm. The report includes the following:

- Analysis of the IIT/D use case with examples.
- A design of the use case.
- Code to produce the graph based design based on neo4j as per the reference code provided in the Build-a-thon repo.

1 References

[FGAN-use cases] ITU-T Focus Group Autonomous Networks Technical Specification “Use cases for Autonomous Networks”
https://www.itu.int/en/ITU-T/focusgroups/an/Documents/Use-case-AN.pdf

[Build-a-thon 2022] https://github.com/vrra/FGAN-Build-a-thon-2022

[FG AN Arch framework] Architecture framework for Autonomous Networks, https://www.itu.int/en/ITU-T/focusgroups/an/Documents/Architecture-AN.pdf

[TCS Smartmachine team report presentation] https://github.com/viprob-ai/Build-a-thon-2022-TCS-Smart-Machine

[IITD landing page] https://bhartischool.iitd.ac.in/build_a_thon/index.html

2 Abbreviations and acronyms

This document uses the following abbreviations and acronyms:

AN Autonomous Networks

API Application Programmer Interface

SDK Software Development Kit

3 Conventions

None

4 Introduction

Problem Statement I (Slip Detection)

- Given: Object being held by Robotic Hand.
- The object may tend to slip from the grip (by introducing weight change to object).
- This task will have constant simulated real world constraints like gravity.
- Overall this task requires an estimation of the force to be applied on the object for a successful grasp.
- It is desirable to leverage latency-aware learning technique to
  - Find if the object is about to slip
  - Applying the appropriate control to prevent the slip
    - Without breaking or deforming the object (force estimation)

- **Dataset-Description:**

Allegro Hand : 16 – Joint Readouts

Raw Data Features :

1. Joint Force,
2. Joint Position
3. Mass
4. Size

Raw Data Format:

N x [16-Joint Force,16-Joint Position, 1 Mass, 1 Size] = N x 34 Feature - Data set.

N= Number of recorded time-steps.

Problem Statement II (Object Detection)

- Given: Object currently held by the Robotic Hand.

- The Robotic Hand records its current states (in the form of angular configuration and forces exerted on its joints)
- Using this modality the task is to detect the shape of the object.

- **Dataset-Description:**

Allegro Hand : 16 – Joint Readouts

Raw Data Features :

1. Joint Force,
2. Joint Position,
3. Mass

Raw Data Format:

N x [16-Joint Force,16-Joint Position, 1 Mass] = N x 33 Features - Data set.

N= Number of recorded time-steps.

5 Design

<u>Problem Statement I (Slip Detection)</u>

- **Feature Engineering**

Model Features Space:

1. Joint Force
2. Joint Position,
3. Force Derivative,
4. Position Derivative

The target label for slip event and crumple event are transformed into 4 possible combined classes for the model to predict the slip and crumple state at any given time step as a combined output.

Target Classes:

1. no-slip and no-crumple
2. slip but no-crumple
3. no-slip but crumple
4. slip and crumple

Any combination of above individual features forms the basis of the feature set for the model to train upon.

All individual features and possible combinations were fed to the learning model during experimentation.

- **Methodology**

For the defined problem statement of detecting the slip event as well as crumple event during the grasp-lift phase of object picking , the time-series readouts data from the 16 Joints of gripper is transformed into the required input format of the LSTM based model.

The data is transformed by applying a sliding window to the time series data set with window size equal to the number of previous observations before the model predicts the output for next time_step.

The transformed dataset is shuffled to avoid biased learning.

Input shape : (n_samples,n_previous_time_steps,n_features)

The build model was tried on different features combinations (n_features : [16 - ,16 - ,16 - ,16 -])

output_shape: (n_samples,n_output)

The model predicts a class out of n_outputs (one – hot encoded) classes (n_outputs : [0,1,2,3]) corresponding to each transformed target class.

Problem Statement II (Object Detection)

- **Feature Engineering**

Model Features Space:

1. Joint Force,
2. Joint Position,
3. Mass
4. Joint Force x Joint Position

The target labels can be segregated into unique 13 object type (class: unique shapes and sizes,properties) as well as 6 unique object categories(same shape objects).

Classes: 13 Objects class
Categories : 6 Objects categories.

The data-set is pre-processed further based on the unique categories to get the extracted data-set which is then fed to the classifier.

Any combination of above individual features forms the basis of the feature set for the model to train upon.

All individual features and possible combinations were fed to the learning model during experimentation.

- **Methodology**

Class Imbalance : The raw dataset with unique object classes was processed to get the balanced dataset as to avoid the biased learning of the model due imbalanced samples of data of individual classes.

The train dataset is generated by extracting each class dataset equivalent to the category which has the least sample in the raw dataset.

Input data : (n_samples,n_features)

The build model was tried on different features combinations (n_features : [16 - ,16 - ,1 - ,16 -])

output_shape: (n_samples,n_output)

The model predicts a class out of n_outputs (one – hot encoded) classes (n_outputs : 13 classes) corresponding to each transformed target class.

<<Vishnu: graph design: new actors and relations for IITD use case>>

e.g. ML model, robotic arm, edge computing system, Sandbox, controller (robotic arm).

- "ML model" "infers" "slip"
- Controller inputs inference
- Robotic arm applies control message
- Edge computing system hosts the controller
- Sandbox hosts simulation
- Simulation validates the ML model.

6 Code and demo

<<TBD: add the model architecture>>

<u>Problem Statement I (Slip Detection)</u>

Classifier : Random Forest

<u>Problem Statement II (Object Detection)</u>

Model : LSTM

<<training results: table: charts>>

7 Summary

This document provides a report on the analysis of IIT/D use cases, Slip Detection (and Force Estimation)” and “Object Detection” in a robotic arm. The report includes the following:

- Analysis of the IIT/D use case with examples.
- A design of the use case.
- Code to produce the graph based design based on neo4j as per the reference code provided in the Build-a-thon repo.