NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

FYP Project Plan

Name: Sherwin Samson

Matric Number: U2020911J

TITLE: State-of-the-art AI integration methodologies & framework.

SUMMARY:

Currently, the integration of AI components is complex and requires ad-hoc developments that are error-prone and repetitive. The result can be systems that do not work as expected as an integrated whole. Indeed, it is not enough for the individual pieces to be functional in a software system. It is also key that their interfaces are well-defined, that components can communicate consistently for the integrated system to function as expected. For example, if one sub-system provides an unexpected input to the next system, that may lead to wrong behaviours and potential catastrophic failure.

We aim to evaluate the state-of-the-art of framework and methods available for the integration of AI systems and develop a simple application applying them. We will consider application in the domain of Hybrid AI (also physics-inspired AI) for Smart Cities.

IMPLEMENTATION PLAN & GOALS:

(1) As an initial exercise, we start by implementing the Spring System Model and a predictive model that is trained to provide an estimate for the stiffness of springs (k) with a given displacement (U) and force (F) applied to the system. We then evaluate the connectivity of executing these models as separate FMU units as opposed to a complete end to end model.

(2) Next, we reproduce and build upon an Automatic License Plate Recognition (ALPR) model using Python, YOLOv8 (You Only Look Once – Deep learning Object detector) and OCR (Optical Character Recognition) for Singapore's context. After detecting the license plate as text, we use Natural Language Processing (NLTK) techniques to classify them according to:

1. Type of vehicle (Goods Vehicle/ Passenger Car/ Taxis, etc.)
2. Organisation wise classifications:
    a. Law Enforcement
    b. Foreign Diplomats
    c. Military organisations.
    d. Public Buses.
3. Date of license plate issuance.

➔ **Introduction of the Spring System Model:**

The system of spring is composed of a series of springs with a certain stiffness K. When a force F is applied, these springs show a displacement U.

First, we develop a physics model (input K, output U, K = f(U)). Then, using the data from the physics model (and including perturbations to make it more realistic), we train a data model to solve the inverse problem (input U, output K, K = f–1(U)). See Figure 1.

This is intended as an illustration of the Hybrid AI approach. Hybrid AI refers to systems that blend aspects of data-based (traditional black-box AI) with aspects of rule-based systems (logic or physics models). In particular, here we use a physics model as a surrogate/replacement of the real system that we do not have access to.

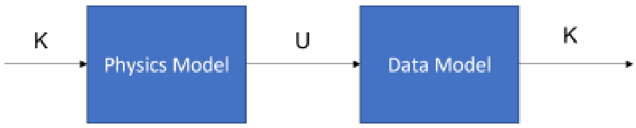

Figure 1: Model pipeline

We consider a system of springs subject to Hooke's law. Hooke's law states that, for a spring with one end attached to a fixed object and another free end that is subject to some force F [2]. If u is the length of extension/compression of the spring from its original position at rest and k is the spring stiffness, we have that F = ku.

APPLICATION & USE CASES:

This application aims to resolve or improve various smart cities' use cases such as:

1. Surveillance/ Security.
2. Law enforcement.
3. Toll charges.
4. Tracking the use of Off-Peak Car Scheme (OPC) in Singapore
    a. Weekend Car (WEC).
    b. Off-Peak Car (OPC).
    c. Revised Off-Peak Car (ROPC).
5. Tracking the use of Vintage and Classic Cars in Singapore:
    a. Vintage (Normal) Vehicle Scheme
    b. Vintage (Restricted) Vehicle Scheme
    c. Revised Vintage Vehicle Scheme
6. Traffic /Parking offences.
7. Traffic Management
8. Route planning: by calculating population density for a given area in terms of the type of vehicles.
9. Correlations between the newly registered vs older license plates.

We can also explore further relevance and use cases of this data for possible problems.

TECHNIQUES:

1. Predictive model (Regression, Neural Network, SVM).
2. Computer Vision (Object detection, OCR (Optical Character Recognition).
3. Natural Language Processing (Classification).

MODELS TO EXPLORE:

1. YOLOv8
2. OpenCV
3. OpenALPR

DATASETS:

4. Open Images V7: Vehicle Registration plate
5. UFPR-ALPR
6. RodoSol-ALPR
7. Number Plate + BoundingBox (India, EU, Brazil, US): Kaggle
8. Car License Plate Detection Dataset: Kaggle

IMPLEMENTATION:

1. Spring System and Predictive Model:
    a. FMU for Spring System.
    b. Predictor Model for Spring System:
        i. Linear Regression
        ii. Neural Network
        iii. SVM
    c. Varying types of noise (Gaussian, Fourier, Noisy Fourier) and predictive models used.
    d. Feature transformation (Normalization, etc.)
    e. Feature extraction (PCA to represent the maximum explained variance, etc.).
    f. FMU for Predictor (Regression, NN, SVM).
    g. Combining both FMUs for Spring System & Predictor models.
2. Automatic License Plate Recognition (ALPR) model:
    a. Using Yolov8 & Optical Character Recognition to detect Car license plates.
    b. Text-based classification of the extracted license plate.
    c. Implement the model as an:
        i. End-to-end model
        ii. Combination of 2 separate units.
3. Evaluate the differences between end-to-end vs non-end-to-end models:
    a. Difficulties.
    b. Accuracies.
    c. Overheads of FMU (if any).
    d. Performance costs.

EVALUATION METRICS:

1. Development of Spring System model and the Predictor as an FMU
2. Evaluation of Spring System model & predictor models end-to-end & non-end-to-end.
3. Development of Multi-Modal AI end-to-end model.
4. Development of Multi-Modal AI model as separate integrated units.
5. Evaluation of end-to-end vs non-end-to-end models.

TIMELINE:

August	1. Initial literature review. 2. Understand FMU/FMI and other tools. 3. Application of FMU/FMI to the Spring System.
September	1. First FMU/FMI demonstrator. 2. Linear Regression Predictor Model for Spring System's equation. 3. Feature transformation (Normalization, etc.) 4. Feature extraction (PCA to represent the maximum explained variance, etc.).
October	1. Neural Network/ SVM Predictor Model for Spring System's equation. 2. Varying types of noise (Gaussian, Fourier, Noisy Fourier) and predictive model used. 3. FMU for Predictor (Regression, NN, SVM).
November	1. Combining both FMUs for Spring System & Predictor model. 2. Compare performance, usability, and accuracy for monolithic vs non-monolithic approaches.

December	1. Implement Multi-Modal AI model (Computer Vision and NLP):
	    a. Car Number plate detection then classifying as text.
	    b. Image -> Localisation of Number plate in the Image -> NLP

January	1. Implement the model as a:
	    a. End-to-end model
	    b. Combination of 2 separate units.
	2. **Submission of Interim Report**

February	1. Evaluating end-to-end vs non-end-to-end models
	    a. Difficulties
	    b. Accuracies.
	    c. Overheads of FMU (if any).
	    d. Performance costs.

March	**Submission of Report**
April	**Submission of amended Final Report**
May	**Oral Presentation**

RELEVANT LITERATURE AND CODE REPOSITORIES:

Research project	https://descartes.cnrsatcreate.cnrs.fr/ https://descartes.cnrsatcreate.cnrs.fr/wp9-builder/
FMI + NN	https://ecp.ep.liu.se/index.php/modelica/article/view/207/167 https://github.com/modelon-community/PyFMI https://www.scs-europe.net/dlib/2020/ecms2020acceptedpapers/0235_ocms_ecms2020_0054.pdf https://github.com/NTNU-IHB/PythonFMU/
ALPR	• OpenALPR: https://github.com/openalpr/openalpr • OpenCV & ALPR: https://www.folio3.ai/blog/opencv-anpr-recognition-anpr-with-python/ • OpenCV & ALPR: https://pyimagesearch.com/2020/09/21/opencv-automatic-license-number-plate-recognition-anpr-with-python/ • https://dr.ntu.edu.sg/handle/10356/162910 • https://dr.ntu.edu.sg/handle/10356/17956 • https://www.geeksforgeeks.org/detect-and-recognize-car-license-plate-from-a-video-in-real-time/ • Recognizing License Plates in Real-Time: XuewenYang, XinWang: chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://arxiv.org/pdf/1906.04376.pdf • License plate recognition based on edge histogram analysis and classifier ensemble: https://ieeexplore.ieee.org/abstract/document/7422310
Datasets	• Open Image v7: https://storage.googleapis.com/openimages/web/index.html • https://paperswithcode.com/dataset/ufpr-alpr • RodoSol-ALPR: https://paperswithcode.com/dataset/rodosol-alpr • Number Plate + BoundingBox (India, EU, Brazil, US): https://www.kaggle.com/datasets/scholngusmaximus/numberplate-bounding-box-india-eu-brazil-us/code • https://www.kaggle.com/datasets/andrewmvd/car-plate-detection

	• https://datasetsearch.research.google.com/search?ref=TDJjdk1URnlZek0zYm14b2R3PT0sTDJjdk1URnpZMjFpYkRjNWRBPT0sTDJjdk1URnljak15TVRock5RPT0%3D&query=openalpr&docid=L2cvMTFzY21ibDc5dA%3D%3D
Hybrid AI vs. Hybrid Modeling vs. Hybrid Co-simulation vs. Hybrid Twins	• Willard, Jared et al. (2020). "Integrating Physics-Based Modeling with Machine Learning: A Survey". In: arXiv: 2003 . 04919 [physics.comp-ph]. • Rai, Rahul and Chandan K. Sahu (2020). "Driven by Data or Derived Through Physics? A Review of Hybrid Physics Guided Machine Learning Techniques With Cyber-Physical System (CPS) Focus". In: IEEE Access 8, pp. 71050–71073. DOI: 10.1109/ACCESS.2020.2987324. Raissi, M., P. Perdikaris, and G.E. Karniadakis (2019). • "Physics informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations". In: Journal of Computational Physics 378, pp. 686–707. ISSN: 0021-9991. DOI: https : / / doi . org / 10 . 1016 / j . jcp . 2018 . 10 . 045. URL: https : / / www. sciencedirect.com/science/article/pii/S0021999118307125. • Rosafalco, Luca, et al. "Online structural health monitoring by model order reduction and deep learning algorithms." *Computers & Structures* 255 (2021): 106604. • Francisco Chinesta et al. "Virtual, Digital and Hybrid Twins: A New Paradigm in Data-Based Engineering and Engineered Data". In: Archives of Computational Methods in Engineering 27.1 (Jan. 2020), pp. 105–134. issn: 1134-3060, 1886-1784. doi: 10.1007/s11831-018-9301-4. url: http://link.springer.com/10.1007/s11831-018-9301-4 (visited on 06/27/2022). • Hooke's Law — Description & Equation — Britannica. Apr. 25, 2023. url: https://www.britannica.com/science/Hookes-law (visited on 04/26/2023). • Palmes, P.P. et al. (2021) Designing machine learning pipeline toolkit for AutoML surrogate modeling optimization, arXiv.org. Available at: https://arxiv.org/abs/2107.01253 (Accessed: 31 August 2023).