**The Project**

Deep learning models, usually based on deep neural networks, are highly prominent at the moment and being extensively used both in academia and industry for prediction tasks such as image classification, image captioning, and question answering. But we have also begun to observe errors, typically detected when encountering one or more counterexamples: inputs that, when slightly modified, are no longer correctly predicted by the model (e.g., a rotated image, or an image with an imperceptible perturbation of its pixel values). A good software development practice is to introduce a test case whenever an error is found and fixed. Put together, these test cases create a “safety net” for users. However, developers still lack *a comprehensive test framework to systematically detect and characterise errors in deep learning models*.

Deep learning models use known input-output pairs to learn (millions of) parameters so that, layer by layer, they can represent inputs in such a way that each input class is separable from one another in the output space. They err when the learned representation of a perturbed input falls outside the decision boundaries learned for its input class. The problem of error testing is to define a set of test cases based on the model’s deployment requirements and architecture design to assert that the system behaves as specified. This set should be considered sufficient when it includes tests on each decision boundary and inputs on each side of each boundary for each class of inputs.

The overall goal of this PhD project is to streamline the specification and evaluation of well- formed test cases for deep learning models that can be easily fine-tuned to a particular deployment environment and model architecture with just a few parameters. There are three main challenges in this project.

***Global probabilistic guarantees****.* Consider that we are given a trained model and the data used to train and evaluate it; our task is to check for errors. We could use formal neural network verification techniques to test that, for a given input, the model is robust to all perturbations within bounds. Suppose that the model passes the test. In other words, we have proven that all bounded perturbations of that particular input do not affect the model’s prediction. How can we ensure that this local property holds for all inputs in that class or, globally, for each input class? The idea is to provide probabilistic guarantees about the overall correctness of the model by efficiently sampling our input data distribution. Samples must be part of the input distribution, they must be diverse enough to capture all modes of the data distribution, and they must generalise beyond the training data. One way forward, for example, is to learn a generative model, such as variational auto-encoders, for generating inputs. As with traditional software testing, the challenge remains to ensure high test coverage and do so in a scalable way.

***Error summarisation.***Suppose that our model fails to pass the aforementioned local robustness test. In other words, we have proven that there is at least one input perturbation that causes the model to mispredict it. Which counterexample(s) to report? Within the bounded input region considered, there can be more than one counterexample; and different counterexamples may cause different misclassifications depending on which decision boundary they cross, thus representing different errors. We want to devise a way to systematically navigate the bounded input region, characterise errors found therein, and return a summary.

***Property-based test generation****.* Different model architectures use different learning algorithms (e.g., convolution, or attention) to accomplish a prediction task. Robustness to adversarial inputs is a desirable property for all of them, but robustness criteria depend on the task’s deployment requirements (e.g., robustness to semantic perturbations such as rotation, scale, and brightness for image perception models), as well as the inductive bias(es) of learning algorithms (e.g., translation equivariance for convolution, or permutation equivariance for attention). We want to devise a way for developers to specify properties to verify and then automatically generate tests from them so as to cover all relevant test cases.

**Research Areas:**

**Error Detection and Characterization:**

Investigate methods to identify and characterize errors caused by perturbations in inputs.

Explore techniques to systematically test deep learning models against a variety of input modifications.

**Global Probabilistic Guarantees:**

Develop methods to ensure that robustness to perturbations for a given input extends to all inputs in a class or globally.

Consider the use of generative models for input generation and efficient sampling of input data distribution.

**Error Summarization**:

Create methodologies for summarizing errors in deep learning models, particularly those that fail local robustness tests.

Develop systems to navigate bounded input regions and characterize the errors found.

**Property-Based Test Generation:**

Focus on specifying properties for different model architectures and learning algorithms.

Automate the generation of tests based on these specified properties to cover all relevant test cases.

**Methodological Approach:**

**Literature Review:** Start with a comprehensive review of existing test frameworks, error detection methods, and robustness criteria in deep learning.

**Experimental Design:** Conduct experiments to test various hypotheses related to model robustness and error characterization.

**Data Analysis and Model Development:** Analyze experimental results to develop or improve testing frameworks.

**Generative Modeling:** Explore the use of generative models like variational auto-encoders to create diverse and representative test cases.

**Ultimate Goal:**

The ultimate goal is to establish a robust, adaptable testing framework for deep learning models that:

Effectively detects and characterizes errors.

Provides global guarantees on model robustness.

Simplifies the process of test case generation based on model properties and deployment requirements.

**Challenges and Considerations:**

**Scalability**: Ensure that the testing framework can handle large-scale models and datasets.

**Diversity of Test Cases:** Generate test cases that cover a wide range of scenarios and input perturbations.

**Interpreting Error Summaries:** Develop a systematic approach to interpret and utilize the error summaries for model improvement.

**Balancing Specificity and Generality**: Straddle the line between creating tests specific enough to be meaningful and general enough to be widely applicable.

**This project presents a sophisticated challenge in the field of deep learning model testing and evaluation. Let's break down the key components and research questions:**

1. **Understanding the Problem Space:**
   * **Deep learning models are prone to errors when encountering slightly modified inputs (e.g., rotated images, imperceptibly perturbed images).**
   * **Current test frameworks for deep learning models are insufficient for systematic error detection and characterization.**
2. **Primary Objectives:**
   * **Develop a comprehensive test framework for deep learning models.**
   * **Ensure the framework is adaptable to various deployment environments and model architectures.**
3. **Main Challenges:**
   * **Global Probabilistic Guarantees: How to ensure a local property (robustness to perturbations for a given input) holds globally for each input class.**
   * **Error Summarization: Systematically navigate the bounded input region, characterize errors, and return a summary of these errors.**
   * **Property-Based Test Generation: Develop a method for specifying properties to verify and automatically generate tests covering all relevant cases.**

**Potential Research Questions:**

1. **Global Probabilistic Guarantees**:
   * How can we efficiently sample from an input data distribution to provide probabilistic guarantees about a deep learning model’s overall correctness?
   * What are the best practices for achieving high test coverage in deep learning models while ensuring scalability?
2. **Error Summarization**:
   * What methodologies can be developed to systematically explore the bounded input region and effectively characterize and summarize errors in deep learning models?
   * How can these methodologies differentiate between types of errors based on the decision boundaries crossed?
3. **Property-Based Test Generation**:
   * How can a framework be designed for developers to specify verification properties and automatically generate robust tests for different deep learning model architectures?
   * What are the unique challenges in creating robustness criteria tailored to specific tasks (e.g., image perception models) and learning algorithms (e.g., convolution, attention)?

Based on the recent research, here are some current test frameworks and approaches for deep learning models that address systematic error detection and characterization:

**Audee: Automated Testing for Deep Learning Frameworks** by Q. Guo, X. Xie, Y. Li, et al. (2020): This framework focuses on the underlying frameworks on which all deep learning (DL) models depend. It defines a test case for DL framework testing, addressing the need for comprehensive testing in deep learning development​​.

**An Empirical Review of Deep Learning Frameworks for Change Detection**: Model Design, Experimental Frameworks, Challenges, and Research Needs by M. Mandal, S.K. Vipparthi (2021): This study presents an empirical review of milestone deep learning algorithms and datasets for change detection, which can be a vital aspect of testing frameworks​​.

**Is Using Deep Learning Frameworks Free? Characterizing Technical Debt in Deep Learning Frameworks** by J. Liu, Q. Huang, X. Xia, et al. (2020): This paper explores the concept of 'technical debt' in deep learning frameworks and addresses the need for more robust testing practices​​.

**Toward Understanding Deep Learning Framework** Bugs by J. Chen, Y. Liang, Q. Shen, et al. (2023): This research focuses on understanding bugs in deep learning frameworks and measures test coverage, offering insights into potential improvements in testing methodologies​​.

**DLBench**: A Comprehensive Experimental Evaluation of Deep Learning Frameworks by R. Elshawi, A. Wahab, A. Barnawi, et al. (2021): This paper presents a comprehensive evaluation of different deep learning frameworks, considering design considerations, metrics, and main features under test​​.