Analyzing Missed Optimization Opportunities in Dynamic Compilers

Compiler writers are often unsure whether an application could be optimized further or not. For example: 1) it is unclear whether the inliner managed to inline all the performance-critical parts of the code, which would allow, e.g., loop fusion or constant folding; 2) it is unclear whether read-write elimination managed to restore sufficient type information to trigger additional optimizations; 3) it is unclear whether tail-duplication duplicated all critical parts of the code; 4) in a multithreaded program, it is unclear if certain synchronization primitives are necessary, in which case coarsening or removing them can decrease their cost. More precisely, technical reasons for those examples can be that a simplifying transformation designed to simplify the intermediate representation (IR) node graph such that the compiler has more information (e.g., inlining or escape analysis) has failed, or an optimization that works on a given instruction pattern directly (e.g., loop fusion) has failed.

This project aims at identifying patterns that are not optimized in state-of-the-art compilers, using a compiler-agnostic methodology. To achieve this goal, the student will be involved in several activities:

1. Development of profilers tracking memory accesses, behavior of threads, and other side-effects, using DiSL [1] and ShadowVM [2].
2. Identify a set of patterns that typical optimizations apply to (e.g., loop fusion), and use those known patterns to identify potentially optimizable runtime executions.
3. Identify a set of applications exhibiting such patterns, detecting missed optimization opportunities on realistic popular applications.
4. Use the findings to implement new optimizations in the Graal compiler [3], as well as to improve the existing ones.

The project is a unique opportunity for the student to deepen his/her knowledge in the domains of dynamic compilation, compiler optimization, dynamic program analysis, and empirical evaluation, all being important skills for a software engineer. The student will work side-by-side with the members of the Dynamic Analysis Group at USI, and will receive support in learning advanced topics that will strengthen her/his abilities as a computer scientist. Applicants interested in this project should be enrolled in the MSc program, have a good knowledge of Java, C/C++ and UNIX-based operating systems, excellent programming skills, and deep interest in the fields of compilers and dynamic analysis. Previous experience with DiSL, ShadowVM and Graal is a plus.

Responsibles: Prof. Walter Binder <walter.binder@usi.ch> and Dr. Andrea Rosà <andrea_rosa@usi.ch>

References:
Contingency in parallel applications arises from the need of accessing shared mutable resources (such as thread-safe data structures) concurrently. An excessive amount of contention is known to significantly impair performance in parallel applications. This project aims at profiling and analyzing the contention caused by threads accessing shared resources, locating situations where contention causes performance issues.

This project is composed of three main tasks:

1. **Selection of key metrics characterizing contention in parallel applications.**
   We will select key metric to quantify the amount of contention in a parallel application and locate situations where excessive contention leads to suboptimal performance. To this end, we will focus on metrics from different layers of the system stack (i.e., framework, runtime, OS, and hardware).

2. **Design and development of a new methodology to accurately profile contention metrics.**
   We will develop efficient instrumentations and data structures for collecting contention metrics accurately and with low measurement perturbation throughout the system stack. We will use DiSL [1] and ShadowVM [2] to implement efficient contention profilers.

3. **Performance characterization.**
   We will use the new profilers to analyze parallel workloads, characterizing the contention experienced by these applications and identifying those suffering from excessive contention.

The project is a unique opportunity for the student to deepen his/her knowledge in the domains of concurrent programming, dynamic program analysis, and empirical evaluation, all being important skills for a software engineer. The student will work side-by-side with the members of the Dynamic Analysis Group at USI, and will receive support in learning advanced topics that will strengthen his/her abilities as a computer scientist. Applicants interested in this project should have a good knowledge of Java, C/C++ and UNIX-based operating systems, excellent programming skills, and deep interest in the field of concurrent programming and dynamic analysis.

**Responsibles:** Prof. Walter Binder &lt;walter.binder@usi.ch&gt; and Dr. Andrea Rosà &lt;andrea.rosa@usi.ch&gt;

**References:**


Calling for a project

Visualizing a Digital Library Interface

About the project:
The project is based on the work of (Ruotsalo et al. 2013 and Nandan 2015). The system is basically based on the paradigm of the interactive intent modeling that enables the users to interact and assess the search results timely. The interface is visualized by implementing a radial layout, and the interface is divided into two sections: the radar displays where relevant intents are closer to the center, and the search result list. The search results will be updated once the user change the intents in the layout according to the relevance of the results.

Aim of the project:
The project aims to provide users of digital libraries with an alternative interactive UI instead of the traditional federated search interface. The ultimate goal is to facilitate user interactions with digital libraries and improve the UX.

The architecture (short description):
The project is a web-based project based on the paradigm of the Radial layout. The project consists of two phases: back-end and front-end phase. The back-end phase includes creating the dataset, indexing, implementing machine learning techniques, and the interface design. Basically, the interface consists of (a) a query box, (b) the result box and (c) an interactive Intent Radar as an interactive area.

The skills required for this project:
A full stack developer with an adequate knowledge of machine learning techniques is a perfect candidate for such project.

References:

**UROP project:** Python front-end for *Utopia*, a C++ library for parallel scientific computing

**Abstract**

*Utopia* is an open source C++ library (bitbucket.org/zulianp/utopia) developed at ICS-USI in collaboration with CSCS (Swiss National Supercomputing Centre). The goal of this library is to provide hardware portable (CPU/GPU) codes for parallel algebra which can be employed for scientific simulations, optimization problems, and machine learning algorithms. Utopia is currently being used in several PASC (pasc-ch.org) projects such as FASTER, AV-Flow, and StagBL.

The goal of this project is to create a front-end interface for *Utopia* using SWIG (swig.org) which stands for Simplified Wrapper and Interface Generator. SWIG allows to create interfaces between C/C++ with a multitude of languages (e.g., Python, Java, Javascript, Octave, …) hence enabling *Utopia* to be employed with any one of them.

The project focus will be on API (Application Programming Interface) design, C++ and Python programming.

During the course of this project the student will:
• work independently on specific development goals,
• research the state of the art in scientific software,
• participate in developing open-source community codes,
• interact with developers and users,
• learn and use basic concepts of C++,
• use the GIT version control system, CMake/Makefile build systems, and Docker containers.
• create an application of their choosing using the produced Python (or any other supported language) interface.

**Prerequisites**

Basic knowledge of C/C++, Object Oriented Programming, and Python.

**Advisors:** Prof. Dr. Rolf Krause, Dr. Patrick Zulian, Alena Kopanicakova

Simulation of a heart valve (AV-Flow)
Contract-aware test-case generation

**Supervisors:** Carlo A. Furia and Andrea Mocci

**Host:** Software Institute/Code Lounge

**Prerequisites:** familiarity with the Java programming language, and some knowledge of unit testing (such as with JUnit).

**Student target:** third-year bachelor students, or master’s students.

Tests are invaluable to help programmers improve the correctness and quality of the software they maintain. As a result, writing tests has long become a standard practice of software development, and even *automatic* test-case generation tools — such as Randoop [1] and EvoSuite [2] — have become increasingly scalable and effective. However, tests are only as good as the specification they check. While manually written tests can encode, in principle, arbitrary complex specifications as assertions, automatically generated tests can only check for basic conditions such as crashes and uncaught exceptions. This limits the variety of tests they can generate, and hence their effectiveness as bug-finding tools.

In this project, you will extend the process of automatic test-case generation tools so that it can use information about the expected behavior of the system under test. The basic idea is that, given a class under test and its specification, the extended test-case generator can automatically produce inputs that *test the specification* and are capable of triggering buggy behavior that deviates from it. The central part of the project will extend an existing test-case generator tool by specializing its *generation algorithm* so that it can use the specification to classify generated tests into passing and failing, and to guide the generation of new tests that cover new program behaviors.

The system’s specification will be encoded as *contracts* — assertions that should hold in different parts of a program. Contracts are added to a Java class using the jSicko framework [3]. Since Java programs do not usually come with contracts but only informal specifications (such as JavaDoc method summaries), part of the project will consist in developing contracts for a number of Java classes and libraries (possibly adapting them from other languages [4]).

As test-case generation tools, the project will consider both Randoop and EvoSuite. These are state-of-the art tools that work on real-world Java programs and are quite effective. The two tools perform the same task (test-case generation) using quite different techniques: Randoop uses random generation, whereas EvoSuite use genetic algorithms. Targeting both will allow you to try out different approaches and ideas, and to understand which test-case generation technique works better under what conditions.

[1] [https://randoop.github.io/randoop/](https://randoop.github.io/randoop/)
[3] [https://github.com/si-codelounge/jsicko-tutorials/](https://github.com/si-codelounge/jsicko-tutorials/)
[4] [https://bugcounting.net/publications.html#fm14-coat](https://bugcounting.net/publications.html#fm14-coat)
Pest insects (e.g., *Popillia japonica*) feed on leaves, flowers and fruits plants, such as apple, drupaceous, grapevine, corn, blueberry, raspberry and other agricultural species. Some of these plants are widespread in Ticino and important for its agriculture. Pest insects are different one another, but some of them are similar to other insects that are not dangerous for the local environment. It has been shown that it is important to find and eradicate as early as possible pest insects when they appears in certain locations since eradication at later stages becomes ineffective and extremely expensive. For this reason, Cantonal Authorities are going to start using a software application that will allow citizens to report the presence of specific pest insects. Report from citizens need to be verified by checking the uploaded pictures of the insect and, if required, proper countermeasures need to be taken against the insect. At the moment, there is no automatic way of verifying the presence of pest insects in images and, therefore, human operators need to perform manual inspection. This prevents the system from providing immediate feedback to the citizens and slows down the action of the authorities. Thus, a system for automatic identification of pest insects would greatly enhance usability and effectiveness of the reporting system.

Goal of this project, which will be in the framework of a broader collaboration with the Agroscope Research Centre Cadenazzo, is to explore the possibility of automatically classifying images of different pest insects by using deep learning methods. The student will have access to a dataset composed of labeled insect images. By using this dataset, the student will explore different methods for the classification of the images of different pest insects, distinguishing them from non pest insects as well as identifying some of the specific insects. Different classifiers will be evaluated based on their accuracy.
An Improved Ear-Training Application for Musicians
Proposal for Undergraduate Research Opportunity Program (UROP 2020)

Supervisor
Dr. Roberto Minelli
Software Institute - USI, Lugano

Co-supervisors
Prof. Nadir Vassena & Alberto Barberis
Scuola Universitaria di Musica-SUPSI,
Conservatorio della Svizzera italiana

ABSTRACT—To develop a better understanding of what they hear, musicians perform a variety of exercises generally known as ear-training or education for listening. Among these exercises, students have to recognize intervals, chords, melodic dictation, etc.

Existing commercial ear-training products are limited. For this reason, Alberto Barberis developed ti ascolto a prototypical tool to support 5 ear-training exercises based on the high-level ear-training class of Maestro Ivo Antognini at the Conservatorio della Svizzera italiana. Its main purpose was to offer a highly customizable ear-training tool suitable for professional musicians. The tool was developed using the Max/MSP visual programming software, a limited environment that offers limited customization possibilities. Besides, the tool was implemented as a desktop application where users have to manually install all the dependencies. This process was both error-prone and time-consuming. To overcome these limitations, Simone Masiero under the supervision of Alberto Barberis and Roberto Minelli, developed SOLO (https://tamburo11.github.io/SOLO) a web-based application acting as a container of exercises for ear-training. As a proof-of-concept, Masiero implemented one of the available ear-training exercises: Melodic Intervals Recognition, where students have to recognize the distance, in terms of pitch, between two consecutive notes.

Music students of the Conservatorio della Svizzera italiana who employed SOLO in their curriculum were very satisfied, which calls for an extension of the tool. The goal of this UROP is to extend SOLO by supporting the following two ear-training exercises:

1. **Harmonic Intervals Recognition.** In this exercise, the student has to recognize the distance, in terms of pitch, between two simultaneous notes.

2. **Chords Recognition.** In this exercise, the student has to recognize between all possible major and minor triads and all possible seventh chords, in all inversions.

For more information, interested students can contact:
- Alberto Barberis <alberto.barberis@conservatorio.ch>
- Roberto Minelli <roberto.minelli@usi.ch>
Luce Research Lab UROP Internship

Do you want to have an impact by contributing to tools and artifacts or by conducting studies that help people, in universities or schools, locally or internationally, more easily learn to program? Then you may want to consider a UROP internship at the Luce research lab with Matthias Hauswirth and his group.

We offer multiple topics for students with all kinds of interests and skills. Projects in our lab span the spectrum from work on program analysis tools that provide educational feedback on code, over highly visual interactive educational web apps, to qualitative human subject research on conceptual understanding, all the way to the production of novel kinds of videos to help educators improve their teaching.

Do you, or would you, like to do any of the following?

- Explain complex concepts or processes to others
- Know a programming language really well
- Develop programs that analyze programs
- Craft highly intuitive user interfaces
- Interview people to probe their understanding
- Design a visual programming language
- Analyze people’s problem-solving approaches
- Produce creative educational videos

If any of the above points triggers your interest, please come talk to us to find out more. We don’t have a canned project for you, but we have lots of ideas and lots of needs, and we would like you to work on something you are truly excited about.

Your work will be connected to two ongoing research projects at Luce: a Swiss National Science Foundation project on Conceptual Change in Learning to Program, and an ACM SIGCSE Special Project on Mastery Learning. While these projects are centered at Luce, they involve collaborations with researchers in Canada, Singapore, Portugal, and Sweden.

To find out more, contact Matthias.Hauswirth@usi.ch.
Sensor-Based Personal Assistant to Promote Self-Learners’ Engagement

UROP project proposal at the Università della Svizzera Italiana (USI), Lugano, Switzerland

Overview of the project

In recent years, mobile and wearable sensing technologies have enabled individuals to unobtrusively and passively monitor their daily lives. Mobile and wearable devices embedded with a wide variety of sensors enable a new generation of digital personal assistants that can actively and accurately monitor, model and promote their users’ health, work performance and overall wellbeing. Digital personal assistants are increasingly becoming digital partners of our everyday computing experiences offering a variety of purposeful information and utility services. Despite the significant amount of research in this field, the deployment of mobile and wearable sensor-based personal assistants in real-life settings is still complex and prone to errors. In this project the main goal is to design and develop a real-time sensor-based personal assistant, which continuously and unobtrusively recognizes the engagement level of the user with the studying activities. The personal assistant should be able to collect data from contextual and physiological sensors embedded in wristband, earbuds and smartphone and use it to infer users’ engagement. The detected engagement level will then be provided as feedback to the user in a timely and effective manner to increase awareness about her engagement level.

Specific tasks

1. **Mobile application** – Implement a mobile application that serves as personal assistant to self-learners to keep track of the duration and type of their studying activities.
2. **Real-time data processing** – Extend the mobile application to process streams of raw sensor data from the wristband, earbuds and smartphone and use them to automatically detect users’ engagement in real-time.
3. **Data visualization and feedback** – Design and implement features in the application to provide interpretable feedback to the user in the form of data visualizations and personalized actions to help self-learners manage their studying activities.

Required skills and knowledge

- Mobile application programming (using Android or a cross-platform programming framework like, e.g., Flutter)
- Data analysis and visualization skills (using Python programming language and Jupyter notebook)

Supervisors and contact information

Shkurta Gashi, shkurta.gashi@usi.ch, Elena Di Lascio, elena.di.lascio@usi.ch, Prof. Dr. Silvia Santini, silvia.santini@usi.ch
Laughter Recognition Using Earbuds

UROP project proposal at the Università della Svizzera Italiana (USI), Lugano. Switzerland

Background

A growing number of pervasive systems integrate emotion recognition capabilities with the aim of improving the interaction with the user or to foster human well-being. Emotion-aware systems collect data from sensors embedded in personal devices – such as smartphones, wristbands, chest bands and other –, recognize users’ emotional state using machine learning techniques and then adapt their behavior accordingly – for example by sending recommendations or in case of virtual agents, changing the vocal tone.

Laughter is a key expression of positive emotions. Enhancing emotion-aware systems with the ability of recognizing laughs can have many applications. A laughter recognition system could be embedded in quantified-self technologies – e.g., through a “laughter counter” – to make users gain awareness of their emotional expressions. Being also a significant sign of appraisal, recognizing laughs can be used in audience-feedback systems and automatically make presenters aware of the audience reactions.

In the last years researchers have proposed several approaches to automatically recognize laughter episodes mainly using data derived from microphones and cameras. More recently researchers have started investigating the possibility of using sensors embedded in wristbands to recognize laughter episodes. In particular physiological data – heart rate and skin conductance – and movement data – derived from the acceleration sensors – have been successfully used. Recent interest is growing in using sensors embedded in earbuds to recognize people’ activities as eating or smiling. Being positioned in the ears earbuds seem to present a suitable and promising solution for recognizing laughs.

The aim of this project is to investigate the feasibility of using sensor data embedded in the earbuds – such as accelerometer and microphone – to recognize laughter episodes.

Required skills and knowledge

- Mobile application programming (using Android or a cross-platform programming framework like, e.g., Flutter)
- Knowledge of Python programming
- Knowledge of data analysis and machine learning techniques.

Expected outcomes

The student is requested to select two of the following steps based on prior knowledge and preferences:

1. Implement a data collection system to gather data from the eSense earbuds.
2. Run a study in laboratory settings to collect data.
3. Implement a machine learning pipeline to recognize laughter episodes using data derived from the earbuds.

Supervisors and contact information

Elena Di Lascio: elena.di.lascio@usi.ch, Shkurta Gashi: shkurta.gashi@usi.ch, Prof. Dr. Silvia Santini, silvia.santini@usi.ch
Developing an Interactive Tool to Allow Human Control in MLS Testing

Paolo Tonella, Michael Weiss
paolo.tonella@usi.ch,Michael.weiss@usi.ch

In this project, you will develop an open-source tool which assists software developers with the integration, mocking and surveillance of the Machine Learning components in their software.

Machine Learning (ML) has gained a lot of attention in software engineering in recent years. Increasingly, ML algorithms are integrated into a Machine Learning based Systems (MLS) where they interact with a large amount of ‘regular’ code. Thanks to modern APIs, integrating ML algorithms into a software system is now as easy as it has never been before. For some common applications, pre-trained ML components can be downloaded and integrated in a MLS without having any knowledge about ML. We can expect this trend to go on - allowing more and more software engineers to use ML in their applications.

ML components during testing
ML components are difficult to debug or to imitate their behavior with mocks, amongst other reasons due to their complex encoded functionality and typically large input spaces. In a system with a ML component, even the debugging and analysis of regular code can thus become a tedious task. In addition to these difficulties, ML is often computationally expensive, making it costly task when running it too often during automated testing procedures.

You will develop an open-source tool allowing to overcome these difficulties.

Goal of the Project
The tool will wrap ML components during testing to allow the tester to override (and analyze) the behavior of the ML component. Specifically, the tool will allow to perform the following operations in a friendly user-interface:

1. Review the inputs and outputs of the ML component.
2. Override / Mock the ML components behavior through a web interface or with values persisted in a database.
3. Report important metrics and properties of the ML components behavior.

To achieve these goals, the following components will have to be implemented:

1. An ML component wrapper (python).
2. An interface using web-technologies for user interactions (javascript or typescript, css, html, python).
3. A cache (database), storing previous human inputs (sqlite).
4. A config file allowing to customize the tool functions.

This is also shown in Figure 1. The specific technologies and functions are of course also dependent on your preferences and may thus change. We will set up more precise requirements at the beginning of your project and adapt them in an agile way during the project.

Prerequisites
We are looking for a student who is passionate about coding, cares about robust, maintainable and beautiful code and who is motivated to contribute his own ideas to the project. You should be proficient in at least one object-oriented programming language.

You do not have to know all the programming languages listed above as learning them is considered part of the project. Similarly, no knowledge of ML is required.

What’s in it for you?
Besides the financial compensation, this project is a substantial addition to the CV of any future software engineer: After the project is successfully completed, you will have contributed to an open-source library. You will then also be able to show practical experience in web-development, machine learning and testing, which are skills recruiters are always looking for.

During the project, we will work closely with you and provide assistance wherever needed. We also always make sure to select and design your work in a way which suits your preferences, wherever possible.

Further information
Do you think you might be interested but have open questions? We are happy to discuss the project with you - just send us an email.

You can also find more information about us on our website: https://www.pre-crime.eu/
Summer Project Proposal: Mutation Testing of Deep Drone

Paolo Tonella, Gunel Jahangirova, Nargiz Huminatova
Software Institute@USI
paolo.tonella@usi.ch, gunel.jahangirova@usi.ch, nargiz.huminatova@usi.ch

Deep Drone Racing
Operating in an unknown and constantly changing environment is one of the main challenges that robotics faces. The Robotics and Perception Group from ETH Zurich addressed this challenge by constructing a fully autonomous and stable vision-based drone. Their approach is based on coupling a convolutional neural network (CNN) with a state-of-the-art path-planning and control system. Given raw images of the environment, the CNN outputs the desired waypoint and speed to approach it. Provided this output, the planner component builds an optimal trajectory to reach the destination, which is then realised by a model-based controller that generates a sequence of motor commands for a drone to follow. The acquired solution is not only platform and domain independent, but it can also run fully on-board on a physical drone, after training in simulation [1, 2] (see Figure 1).

![Figure 1: Simulation data used for training (left) and subsequent testing on physical quadrotor (right)](image)

Mutation Testing
Mutation testing is a technique that deliberately seeds faults in the form of small syntactic changes into the program under test to create a set of faulty programs called mutants. The general principle underlying this approach is that faults injected by mutation testing represent mistakes that programmers usually make. Mutation testing aims to assess the quality of a given test suite in terms of its capability to detect faults. For this, the test suite is executed on each of the generated mutants. If the result for a given mutant is different from the result of running the original program, then the mutation is considered as "killed". The ratio of killed mutants to the overall number of generated mutants is called mutation score. The higher the mutation score, the better the test suite.

The example in Figure 2 shows method subtract. It has two mutations: in Mutant 1 the subtraction is replaced with multiplication and in Mutant 2 it is replaced with addition. If our test suite has only test0, none of the two mutations would be killed (as they all return the expected value 0) and the mutation score is 0%. If we add test case test1() to our test suite, then Mutant 1 gets killed and the mutation score becomes 50%. Once we add test case test2(), both mutations get killed and the mutation score achieves its maximum value of 100%.

![Figure 2: Mutation Testing Example](image)

Deep Crime
Deep Learning (DL) has become an integral part of many groundbreaking projects and applications, including robotics. In traditional software systems the decision logic is implemented in the source code. In contrast, the behaviour of a DL system is mostly determined by the training data set, the model architecture and the training program. Hence, these are the major sources of defects for DL systems. Therefore, for mutation testing of such systems we have designed specific mutation operators that inject potential faults into the training data or the training program. They are implemented in DeepCrime – a tool for mutation testing of DL systems based on a set of real faults.

Project Proposal
In the Summer project, the student will learn how to operate a drone in simulation in order to generate training data and how to train the drone model on the collected data. The next step would require the student to apply the DeepCrime tool to study how the behaviour of the model changes after application of specific mutation operators and to provide a detailed analysis on which of the operators are useful for drone testing.

Further Information
The proposed work is part of the Precrime (Self-assessment Oracles for Anticipatory Testing) ERC project. The interested student can find more information about Precrime on the project’s website: https://www.pre-crime.eu/

REFERENCES
Summer Project: Oracle Improvement for Self-Driving Cars

Paolo Tonella, Gunel Jahangirova
Software Institute@USI
paolo.tonella@usi.ch, gunel.jahangirova@usi.ch

A lot of corporations are actively working on self-driving cars, including large companies such as Google, Ford, and Intel. Some of these cars are already operating on public roads, with at least one unfortunate fatality recently on record. Therefore, properly testing self-driving cars to ensure their security, safety, robustness, and correctness is extremely important.

Architecture of a Self-Driving Car

The key component of an autonomous vehicle is the perception module controlled by the underlying Deep Neural Network (DNN). This DNN takes input from different sensors like camera, light detection and ranging sensor (LiDAR), and IR (infrared) sensor, and outputs the steering angle, braking and etc. needed to guide the car safely under current conditions (see Figure 1).

![Figure 1: Architecture of a Self Driving Car](image1)

Test Oracle

In software testing, an oracle refers to a mechanism which determines if a program under test is functioning correctly or not. For self-driving cars the simplest form of an oracle is whether during a simulation the car completes the given track successfully or gets out of its bounds. However, “not crashing” is a weak form of an oracle, as for a high quality of driving the car has to keep up in the centre of the road, see Figure 2. A stronger oracle should compare the car position with the centre of the lane: assert (car_position == centre_lane). However, the output of a DL system may slightly vary in each run due to randomness. Moreover, for the cases where the outputs are floating numbers, it might be hard to precisely define the expected value. Therefore, the DNN output is allowed to be within a restricted range of the oracle: assert (||car_position − centre_lane|| < threshold). The correct identification of the threshold parameter is extremely important. If the allowed range is too loose, the test inputs would fail to check the correctness of the implementation. If the range is too tight, the test inputs would constantly fail and introduce false warnings of failing tests to developers.

Mutation Testing

Mutation testing is a technique that deliberately seeds faults in the form of small syntactic changes into the program under test to create a set of faulty programs called mutants. The general principle underlying this approach is the assumption that the faults injected by mutation testing represent the mistakes that programmers usually make. Unlike traditional software systems, the behaviour of a DL system is mostly determined by its training data set. For this reason, the existing mutation operators for DL systems inject faults mostly into the training data. Some examples of such mutation operators are introducing noise perturbation into the training data, removing parts of the training data, changing the labels of training data into incorrect ones and etc. Such changes create faulty DL systems that might produce wrong predictions. We have an existing set of mutations for Udacity self-driving car which we have obtained by applying 21 different mutation operators.

Mutation testing can be used to assess the quality of an oracle in terms of number of faulty behaviours it is able to detect (called mutation score), but in turn the quality of an oracle depends on the values assigned to its thresholds. E.g., if the threshold parameter for ||car_position − centre_lane|| is set to 0.10, then the oracle may be able to detect the faulty behaviour introduced by the mutant, but it may also raise false alarms because the car position may deviate so much even when driven by the non-faulty (un-mutated) model. By setting the threshold to a smaller value, such as 0.05, the mutation may still be detected, but false alarms could disappear.

Project Proposal

The student will get familiar with the Udacity simulator and our dataset of mutations. For each of the generated mutated version the student should identify whether the mutation causes the car to crash during a simulation in a given track or not. If yes, then this mutation is killed by the weakest oracle (i.e. the crash). For the mutations that do not cause any crash, the optimal value for the threshold parameter will be computed, so as to minimize the false alarm rate and to maximize the mutation score.

Further Information

The proposed work is part of the Precrime (Self-assessment Oracles for Anticipatory Testing) ERC project. The interested student can find more information about Precrime on the project’s website: https://www.pre-crime.eu/
Autonomous self-driving vehicles are just around the corner. Most major manufacturers including Tesla, GM, Ford, Mercedes, BMW, and Waymo/Google are building and actively testing different types of autonomous vehicles. Recent results show that autonomous cars have become very efficient in practice and have already driven millions of miles without any human intervention.

Development of a Self-driving Car

Data collection for the development of autonomous cars typically requires actual vehicles driven by professional drivers during specific hours on some designated roads. This provides detailed sensor data of the vehicle that are recorded, played back, and recreated within a simulator to obtain comprehensive scenarios for the development of a reliable autopilot (Figure 1).

Can we “repair” a DNN?

The goal of the project is to implement a method for accommodating new behaviours in an already trained DNN without retraining it. This is called “repair” because the aim is to improve the effectiveness of the DNN with respect to the desired task in a specific driving condition.

The motivation is not to replace the existing learning policy, but rather to provide a focused, temporary intervention towards a specific part of the DNN.

Tasks.

- identify the part of the DNN (i.e., the weights) responsible for a certain undesirable behaviour;
- devise an algorithm that modify such weights to include the new, desired behaviour accordingly.

Both tasks are related and can be tackled effectively by well-known engineering solutions, namely, feature localization, and search-based input generation.

Project Proposal

The candidate is required to implement a software component to automatically “repair” the weights of a DNN that performs autopilot function. The candidate will learn how to train a DNN-based autonomous driving system and let it operate to devise potential weak spots that can point to improvements. Research will be carried out on the input generation of the weights for an already trained autonomous driving system, and how to improve its behaviour without retraining. Finally, students will be able to verify their development progress on real simulators for self-driving cars available in the research lab.

Further Information

The proposed work is part of the Precrime (Self-assessment Oracles for Anticipatory Testing) ERC project. The interested student can find more information about Precrime on the project’s website:

https://www.pre-crime.eu/
Autonomous self-driving vehicles are just around the corner. Most major manufacturers including Tesla, GM, Ford, Mercedes, BMW, and Waymo/Google are building and actively testing different types of autonomous vehicles. Recent results show that autonomous cars have become very efficient in practice and have already driven millions of miles without any human intervention.

Architecture of a Self-driving Car
The key component of an autonomous vehicle is the perception module controlled by the underlying Deep Neural Network (DNN). The DNN takes input from different sensors like camera, light detection, ranging sensor, and infrared sensor, which measure the environment, and outputs the steering angle or braking commands necessary to operate the car safely (Figure 1).

Simulation Platforms
On-road testing of autonomous cars is typically restricted to a small number of vehicles driven by professional drivers during specific hours on some designated roads with specific speed limits. Such testing is often expensive and time-consuming. It is further impractical to perform a full-fledged on-road vehicle-level testing after every change to self-driving software systems. To ensure safety of self-driving technologies, vehicle-level testing alone is neither enough nor practical. Therefore, it is complemented by testing methods performed on software simulators (see Figure 2).

Simulation-based platforms of autonomous cars are very popular as they allow engineers to run a much larger number of test scenarios compared to vehicle-level testing without being limited by conditions enforced during on-road testing.

Project Proposal
Researchers and AI experts are debating on whether the results obtained on a simulation environment are representative enough of real-world situations. Though, prior to posing such question, we should get a better understanding on the pros, cons, and limitations of existing simulators, and how they compare with each other.

Different simulators exist in the open-source realm. Instances of such simulators are Udacity, BeamNG, and CARLA. Each have different characteristics, and may be suitable for different tasks. The goal of the project is to compare some of the most adopted simulation-based platforms for self-driving cars. A first question the project will answer is: If an engineer needs to perform a certain task (e.g., road generation), what simulator is best suited for that task? This first part of the study will highlight differences and commonalities in the degree of support to the engineers in self-driving cars development and testing.

Secondly, when commonalities are identified, a more in-depth study will be performed to compare how the various simulators are performing the same task. For instance, the same self-driving car module will be executed on the same scenario (i.e., road shape/length) on two different simulators. Hence, the second question the project will answer is: Does autopilots effectiveness change if executed on the same scenario on different simulators?

Tasks.
- familiarize with existing self-driving car simulators, and the self-driving car model.
- highlight the characteristics of each simulator.
- devise a series of tasks that can be performed by at least two simulators.
- experiment the effectiveness of existing autopilots at executing the defined tasks on different simulation platforms.

Further Information
The proposed work is part of the Precrime (Self-assessment Oracles for Anticipatory Testing) ERC project. The interested student can find more information about Precrime on the project’s website:

https://www.pre-crime.eu/
Federated Learning of Human Mobility

UROP project proposal at the Università della Svizzera Italiana (USI), Lugano, Switzerland

Overview of the project

The ubiquity of mobile phones has significantly changed our understanding of human mobility – and consequently our use of this knowledge in the form of mobility models. Due to the density of today's communication infrastructure (dense urban areas may have cell towers every tens of meters), mobile phone operators are able to track a subscriber's path through a city with block-level accuracy. Self-tracked systems that use GPS (e.g., Google Maps) may allow service providers even more fine-grained tracking capabilities. Mobility models allow providers to characterize these traces, e.g., for predicting future activities of individuals to optimize network handovers, or to simulate large-scale network load.

A plethora of data-driven mobility models have been suggested. The key differentiation between them lies in the set of features they use and the type of predictions they are designed to make. However, all of these models are typically based on offline learning, i.e., they ingest a fixed (large) set of mobility traces and use this to train the respective model.

Recently, a novel distributed approach to machine learning has been presented, called Federated Learning (Konecný et al. 2016). The approach allows training a centralized model while leaving all training data distributed over a large set of clients. In each round, each client independently computes an update to the centralized model, which are then aggregated into a new global model. As communication costs are critical in this setup, a highly efficient communication model is used (McMahan et al., 2016). Furthermore, to preserve the privacy of each client's data, all data is sent blinded, i.e., the (honest but curious) server is unable to learn the individual data that each client's model contribution is based on (Segal et al., 2017). Federated Learning is in practical use across a range of Google services, such as predicting words in its mobile keyboard app (Chen et al. 2019). Google's Open Source machine learning framework TensorFlow has been adapted to support this approach, called TensorFlow Federated -- see https://www.tensorflow.org/federated/.

The goal of this project is to explore the use of TensorFlow Federated in order to learn mobility models.

Specific tasks

(1) Understand the basic principles behind TensorFlow Federated
(2) Identify suitable mobility models and data sets.
(3) Develop a proof-of-concept implementation
(4) Plan and perform a small pilot deployment

Required skills and knowledge

O TensorFlow
O Python programming
O Mobile programming a plus

Supervisors and contact information

Prof. Dr. Marc Langheinrich (marc.langheinrich@usi.ch)
Bibliography

Physiological Synchrony and its Effects on Recall in Peer Meetings

UROP project proposal at the Università della Svizzera Italiana (USI), Lugano, Switzerland

Overview of the project

In the past years, technology advances made it possible to use mobile phones and wearable devices as ubiquitous and discreet sensing devices. The rich range of sensors available allows us to continuously measure different types of changes not only in the environment (location, level of noise, lighting conditions), but also in the person using or wearing them (movement, heart-rate, electrodermal activity, EEG). Raw data gathered from sensors can afterwards be used to gain knowledge.

It has been shown that it is feasible to infer the level of attention of a person from the data obtained through a subset of the previously mentioned sensors. In the education environment, for example, this information can be used to promote different teaching strategies, in order to improve student’s engagement, their learning and therefore their education.

There are also systems that aim to augment work productivity, in particular meetings, by recording them and later providing a summary of the topics discussed with actionable items for the next meeting. Physiological information of the participants could be used as an additional input to these systems, as the summary produced by them could be tailored to each participant of the meeting, based on their physiological responses during it.

The goal of this project is to evaluate in a pilot study the effect of physiological synchrony of peers during a meeting, and its effect on memory (posterior recall of events and topics discussed during the meeting).

Required skills and knowledge

- Mobile programming (Android, or cross-platform frameworks)
- Python programming
- Digital signal processing and visualization

Specific tasks

1. Design the study that will be carried out to measure participants' physiological data during a meeting and their posterior recollection of it.
2. Develop the corresponding technical artifacts (mobile application for data collection, automation scripts for preprocessing) to perform the study
3. Conduct a pilot study for data gathering
4. Perform the analysis of the data

Supervisors and contact information

Matías Laporte (matias.laporte@usi.ch), Prof. Dr. Marc Langheinrich (marc.langheinrich@usi.ch)
A new revolutionary product and its commercialization

The project: The project is based on an entrepreneurial idea, proposed by Mr Daniele Cerri (email: daniele.cerri@bluewin.ch), to create a possible new business which could be brought to the mass, world-wide and could be very profitable. The “product” is a modification of an already existing one and is used world-wide by almost everyone (for discretion we omit the real name of the product in this proposal, but you can think of it as a personal device of common use). It is useful in some particular conditions and could be bought in normal shops as well as on internet. Currently the product is just considered useful in specific circumstances, but nothing else. Consumers are not giving too much importance to it, even if they recognize that it is useful. Thus, many consumers are misplacing it, dropping it, forgetting it around, some have even been stolen and others are breaking due to poor quality. This project aims at making the product a desired object, that many people will be interested in buying, owning or even borrowing.

The project has two parts a “business” and a “technical” part. This specific project is concerned with the technical part and will be done in collaboration with another student dealing with the business part (and such collaboration could be tight or loose, at the students’ choice). In particular, this project is concerned with developing the IT infrastructure related to the design and managing of the product, to avoid losing it, misplacing it, but also making it possible to selling it, buying it, personalising it, etc.

Aim of the project: the project will develop the following aspects, focusing on the ones that are most interesting to the student:

- Business model: Complete product’s design and personalization;
- Website/App design and construction: An e-commerce website and/or a mobile phone app where, after logging in, one can find information about the product, configure it (i.e. creating a unique personalised version of the product), buy it, locate it, contact the company (via chat), contact other owners, etc..
- Configurator construction: A configurator on the website that will lead the buyer to preparing his own version of the product;
- Video and communication: After the configuration took place and the product is ready, an animation is launched with a “persona” having specific characteristics (e.g. mimicking the user) who will show off the product just created;
- Localizing technology: A localisation tracker will be inserted in the product to locate a lost product via a website;
- Integration technology: The product could connect with your smartphone to see incoming calls, messages, etc. Additionally, one part of the product could illuminate and/or vibrate when a call/message is received;
- Social Media presence: The ownership of the product will give a owner access to a social media community (via, for example, Facebook, Instagram, or WeChat), where one can discuss about the good and the bad properties of the product, create connections, share it, socialise and (possibly) meet other people that own it, publish photos of owners and users of the product, etc, sharing the all experience of owning and using it.

Skills required for project: A website designer with a good knowledge of programming techniques is a perfect candidate for such project.
Given the commercial sensitivity of the project, full details of the idea will be provided on request. For further information please contact Prof. Fabio Crestani and/or Mr Daniele Cerri (daniele.cerri@bluewin.ch).
Graph Series Analysis of Connected Systems with Geometric Deep Learning

A UROP 2020 project proposal presented by Rolf Krause (USI, ICS), Alessandro Lomi (UIS, ICS), and Lorenzo Bucci (USI, MS Student in AI).

Geometric Deep Learning (GDL) is an extension of Deep Learning (DL) techniques from data defined on Euclidean domains (i.e., images) to graph/manifold structured data. GDL techniques have been shown to achieve very high levels of performance in the analysis of complex systems, with applications spanning biological, physical and social systems. Complex systems are typically constituted by a multitude of agents, whose interactions are often difficult to model or even to observe. Graph theory provides a powerful framework to represent such systems rigorously as graphs. GDL techniques may be used to make inference on the static properties of the underlying interconnections. However, complex systems composed of multiple agents tend to exhibit patterns of collective behavior that change unpredictably over time. This dynamic property of complex systems requires the underlying graph representation to change accordingly in order to track the system’s evolution and predict its future states.

Against this general background, in this project we propose an extension of state-of-the-art GDL models to graph series analysis settings (in a similar way to what Recurrent Neural Networks do in DL contexts). The relevance of such models resides in the need to understand the dynamic rewiring behavior of multi-agent systems. This is particularly important when agents’ connective behavior affect their performance – i.e., their ability to reach individual goals, or collective targets.

We explore the predictive performance of the model in the context of data we have collected on network of patent and paper citations. The structure of these connected citation systems changes as new agents (patents, papers) arrive into the system and modify its global connectivity properties. The contextual objective of the model that we present is to predict the success of papers and patents as measured by the number of citations they receive over time (indegrees). The model will be trained using a citation history dataset, where newly published papers define a citation timeline for previously published papers. Citation timelines, in turn, define patterns attributable to the success of the corresponding papers. Our target is to predict the citation timeline (and hence the success) of newly published papers and patents given their observed pattern of received citations.
Analysis of Image Quality Perception from Brain Waves

Contact: Prof. Piotr Didyk (piotr.didyk@usi.ch), Dr. Okan Tursun (tursuo@usi.ch)
Group website: https://www.pdf.inf.usi.ch/

Motivation

Novel advanced display technologies on desktop and mobile platforms, Virtual/Augmented Reality displays, HDR displays, and other 3D display systems require unique visual optimization strategies for the best viewing experience. As the primary target of these displays, the study of the Human Visual System (HVS) and visual perception has drawn a significant amount of research interest to evaluate the quality provided by new methods and algorithms, and it has a wide range of applications in all stages of image processing. The knowledge gained from the analysis of HVS provides insights into the development of new methods and algorithms, which presents us with the opportunity of understanding the requirements for more ergonomic and entertaining display technologies.

Goal

Given the neural complexity of HVS, the existing research efforts focus on understanding different stages of how our brains process visual information. With the increasing accessibility of different data collection methods such as eye trackers, fMRI, and EEG, we have recently reached a point where we can observe different aspects of HVS in action. Our primary goal in this project is to answer the following question: Can we directly measure the perceived visual quality of an image by observing the brain waves and eye movements of a viewer? For this study, we will use EEG data and gaze positions captured in various observation scenarios with carefully generated images including various types of distortions and conduct statistical analyses on the collected data to understand and find out if there are specific brain waves and eye movement patterns which are correlated with the visual quality of the presented visual stimulus. The findings of this project will shed light on the future evaluation of novel displays and provide an efficient means of benchmarking new display technologies and image processing methods.

Prerequisites

- Good programming skills
- Experience in signal and image processing
- Knowledge of fundamental statistical analysis

Please feel free to contact us for more details