PhD in Information Technology and Electrical Engineering

Università degli Studi di Napoli Federico II

PhD Student: Marco Castelluccio

XXXI Cycle

Training and Research Activities Report – Second Year

Tutor: Carlo Sansone – co-Tutor: Luisa Verdoliva

1. Information

PhD Student: Marco Castelluccio
MS title: Computer Engineering – University of Naples Federico II
PhD cycle: XXXI – ITEE University of Naples Federico II
Fellowship type: PhD student without grant
Tutor: Carlo Sansone – **co-Tutor:** Luisa Verdoliva

I received my MS degree (cum laude) in Computer Engineering from the Università degli Studi di Napoli Federico II in September 2015.

I have been a Software Engineer at Mozilla since November 2015, after two internships and a period of contracting with the same company during my graduate studies. I am a Senior Software Engineer at Mozilla since August 2017.

2. Study and Training activities

Courses

1. "Image Processing for Computer Vision" – Giuseppe Scarpa
2. "Machine Learning" – Carlo Sansone

Seminars

1. Towards Evolutionary Approximate Optimization for Machine Learning – Dr. Yang Yu (Department of Computer Science, Nanjing University, China) – Institute of Electrical and Electronics Engineers (IEEE)
2. Town Hall with Peter Norvig on A.I., Machine Learning, and More – Peter Norvig – Association for Computing Machinery (ACM)

3. Panel and Town Hall: Big Thoughts and Big Questions about Ethics in Artificial Intelligence – Joanna J. Bryson (Associate Professor at University of Bath), Francesca Rossi (Research Scientist at the IBM T.J. Watson Research Centre, Professor at the University of Padova; AAAI & EurAI Fellow), Stuart Russell (Professor at UC Berkeley and Adjunct Professor at UC San Francisco; ACM Fellow, AAAI Fellow, and AAAS Fellow), Michael Wooldridge (Professor at University of Oxford and Research Fellow at Hertford College; ACM Fellow, AAAI Fellow, EURAI Fellow) – Association for Computing Machinery (ACM)
4. Current Trends in High Performance Computing and Challenges for the Future – Jack Dongarra (University of Tennessee and Oak Ridge National Laboratory; ACM Fellow) – Association for Computing Machinery (ACM)
5. Speaking Data: Simple, Functional Programming with Clojure - Paul deGrandis, Lead Developer, Architect, and Director of the Research and Innovation Group at Cognitect – Association for Computing Machinery (ACM)
6. Being agile and lean in constrained and regulated environments - Paul E. McMahon, Principal, PEM Systems – Association for Computing Machinery (ACM)
7. The Next Radical Internet Transformation: How Blockchain Technology is Transforming Business, Governments, Computing, and Security Models – Mark Mueller-Eberstein, CEO & Founder at Adgetec Corporation, Professor at Rutgers University, Senior Research Fellow at QIIR – Association for Computing Machinery (ACM)
8. Software Assessment – David Weiss (Co-founder and Partner, Sustainable Software, LLC), Randy Hackbarth (Co-founder and Partner, Sustainable Software, LLC) – Association for Computing Machinery (ACM)
9. Introducing The Vulnerability History Project – Andy Meneely (Assistant Professor, Rochester Institute of Technology) – Association for Computing Machinery (ACM)

10. Software Quality Measurement – Ravi Sethi (Professor, University of Arizona), John Palframan (Co-founder and Partner, Sustainable Software, LLC) – Association for Computing Machinery (ACM)
11. Preventing Information Leaks with Policy-Agnostic Programming – Jean Yang (Assistant Professor at Carnegie Mellon University) – Association for Computing Machinery (ACM)
12. Application Management with Kubernetes – Kelsey Hightower (Staff Developer Advocate, Google Cloud Platform, Google) – Association for Computing Machinery (ACM)
13. Machine Learning in Neural Representations of Language – Tom Mitchell (E. Fredkin University Professor at Carnegie Mellon University) – Association for Computing Machinery (ACM)
14. Rust training – Florian Gilcher (Asquera GmbH)
15. Monadic Programming for the Web Using React and RxJS – Pat Sissons (Senior Software Developer at Marine Learning Systems) – Association for Computing Machinery (ACM)
16. Breaking New Frontiers in Robotics and Edge Computing with AI – Dustin Franklin (GPU Developer, NVIDIA) – NVIDIA Corporation
17. Three Years in the Startup Trenches: Reflections on People, Product, and Software Evolution – Andrew Ko (Associate Professor at the University of Washington Information School) – Association for Computing Machinery (ACM)
18. AI, People, and the Open World – Eric Horvitz (Technical Fellow and Director of Microsoft Research Labs; ACM Fellow) – Association for Computing Machinery (ACM)
19. Are agile methodologies the right approach for industry - academia research collaboration? – Ivica Crnkovic (Professor at Chalmers University, Mälardalen University) – Association for Computing Machinery (ACM)
20. Open Collaboration, the Eclipse Way – Mike Milinkovich (Executive Director of the Eclipse Foundation) – Association for Computing Machinery (ACM)

21. Lessons Learned From Working In Industry And Academia – Douglas Comer (Distinguished Professor of Computer Science at Purdue University) – Association for Computing Machinery (ACM)
22. ImageNet: Where Have We Been? Where Are We Going? – Fei-Fei Li (Chief Scientist of AI/ML at Google Cloud, Director of Stanford A.I. Lab) – Association for Computing Machinery (ACM)
23. Build Your Next Deep Learning Application for NVIDIA Jetson in MATLAB – Bill Chou (Product Marketing Manager, Mathworks) – NVIDIA Corporation
24. Metamorphic Testing: Introduction and Applications – Sergio Segura (Senior Lecturer at Seville University), Zhi Quan (George) Zhou (Associate Professor at the University of Wollongong) – Association for Computing Machinery (ACM)
25. Algorand: A Better Distributed Ledger – Silvio Micali (Faculty at MIT and ACM A.M. Turing Award Winner) – Association for Computing Machinery (ACM)
26. 55th Crest Open Workshop, “Bimodal Program Analysis” – University College London

External courses

1. Modern C++: C++11 / C++14 / C++17 – KDAB Group

Year	Modules	Seminars	Research	Tot.
1	21 (20)	7 (5)	35 (35)	63 (60)
2	16 (9)	12 (6)	45 (42)	73 (60)
3	(0)	(5)	(55)	60 (60)

Tot.	**37 (30-70)**	**24 (10-30)**	**135 (80-140)**	**180 (180)**

3. Research activity

- **Convolutional Neural Networks for Classification of Remote Sensing Images** – Continued work on using convolutional neural networks for classification of remote sensing images started during the MSc thesis [1], analyzing results on additional datasets [2].

  The results of the work have been published: M. Castelluccio, G. Poggi, C. Sansone, L. Verdoliva – Training Convolutional Neural Networks for Semantic Classification of Remote Sensing Imagery – JURSE2017.

- **Automating the Understanding of Groups of Crash Reports** – Many Software Engineering studies have focused on improving the bucketing of crash reports with different techniques (e.g. by using different distance metrics, like Levenshtein [3], custom stack-based metrics [4,5], by using information retrieval techniques [6,7], etc.).

  The focus of my work is instead on how to automatically describe the buckets' properties in the most useful way for developers. Understanding what makes a crash group meaningfully different than other groups is indeed very often useful for debugging (and in some cases even enough for fixing the crash, e.g. by blocklisting a certain graphics card), but it involves a tedious and error-prone manual exploration of the database of crashes.

  We devised an algorithm, inspired by contrast-set mining algorithms such as STUCCO [8,9] and CIGAR [10], to automatically find statistically significant properties (correlations) in crash groups. Developers used to spend a fair amount of time analysing crash groups, which meant that a) they were not spending their time actually developing a fix for the crash; and b) they might have missed something in their exploration of the crash data (there is a large number of attributes in crash reports and it is hard and error-prone to manually analyse everything). Our algorithm helps developers and release managers understand crash reports more easily and in an automated way, helping in pinpointing the root cause of the crash.

The tool implementing the algorithm has been deployed on Socorro, Mozilla's crash reporting service.

The results of the work have been published: M. Castelluccio, C. Sansone, L. Verdoliva, and G. Poggi – Automatically analyzing groups of crashes for finding correlations – Proceedings of the 2017 11th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering (ESEC/FSE 2017).

- **Emprical Study of Uplifts in Mozilla Firefox** – Collaboration with the École Polytechnique de Montréal on studying uplifts (backports) in the Mozilla Firefox software.

  Mozilla Firefox has adopted a rapid-release model [11,12], called the "train model". In rapid release development processes, patches that fix critical issues, or implement high-value features are often promoted directly from the development channel to a stabilization channel, potentially skipping one or more stabilization channels. This practice is called patch uplift. Patch uplift is risky, because patches that are rushed through the stabilization phase can end up introducing regressions in the code. The aim is to understand the properties of these critical changes vs normal changes; understand which uplifts introduced bugs (by using the SZZ algorithm proposed in [13]) and why; with the ultimate goal of building a model to predict the riskiness of an uplift.

  We have examined patch uplift operations at Mozilla, to identify the characteristics of uplifted patches that introduce regressions, both through statistical and manual analyses, and through interviews with Mozilla release managers. Results show that most patches are uplifted because of a wrong functionality or a crash. Uplifted patches that lead to faults tend to change a higher number of lines of code, and most of the faults are due to semantic or memory errors in the patches. Also, release managers are more inclined to accept patch uplift requests that concern certain specific components, and–or that are submitted by certain specific developers. The results of this study could be used to train machine learners able to recommend patches that are safe to uplift for release management teams.

  The results of this work have been published: M. Castelluccio, L. An, and F. Khomh - Is It Safe to Uplift This Patch? An Empirical Study on Mozilla Firefox - In proceedings of the 33rd International Conference on Software Maintenance and Evolution (ICSME 2017). The paper received a IEEE TCSE Distinguished Paper Award, and we were invited to publish it, with some additions, on the “Empirical Software Engineering” journal.

- **Automatically detecting web-compatibility issues** – Web compatibility is a very important concern for browser vendors. While there are standards describing APIs and behaviors of web browsers, there are often corner cases that are not fully

specified and thus implemented differently. Given the fast pace in browser development, there are also features which are implemented before being standardized and agreed upon between stakeholders. This leads to problems in the field, as website developers sometimes assume the standards are implemented exactly in the same way or because they simply only test with the most widely used browsers, some websites present compatibility problems, working correctly in some browsers but not in others.

Currently, spotting websites which present compatibility problems is pretty much a manual effort. Volunteers and QA professionals manually inspect web sites in different browsers and notify issues on bug tracking systems such as Bugzilla [14] or the Web Compatibility issue tracker [15] (an issue tracker which was built specifically for this kind of problems). Once the issues have been identified, browser vendors can then contact the website authors to notify them of the problems or even suggest fixes. Clearly, the manual process is prone to errors (after inspecting many websites, people can start overlooking differences), very slow and above all it is very far from being exhaustive, since the number of web sites is huge and grows continuously. Moreover, it has to be repeated periodically, since websites are frequently modified.

The purpose of this work is presenting an automatic technique that browser vendors can adopt to quickly spot websites which are not working correctly with their product.

In order to automatically detect web pages that behave differently in different browsers, we propose to use convolutional neural networks (proposed by Fukushima [16] and later refined by LeCun et al. [17]) to evaluate differences in screenshots of page rendering in different browsers. In particular, we are planning to use siamese networks [18], with different training techniques (e.g. finetuning on network pretrained with ImageNet [19]) and different architectures (e.g. VGG [20]).

The dataset that we are building is going to be open to all to reproduce and improve our results.

[1] M. Castelluccio, G. Poggi, C. Sansone, and L. Verdoliva, "Land Use Classification in Remote Sensing Images by Convolutional Neural Networks", arXiv preprint.

[2] G.S. Xia, W. Yang, J. Delon, Y. Gousseau, H. Sun, and H. Maitre, "Structrual High-Resolution Satellite Image Indexing", in Processings of the ISPRS, TC VII Symposium Part A: 100 Years ISPRS—Advancing Remote Sensing Science, Vienna, Austria, 5–7 July 2010.

[3] Tejinder Dhaliwal, Foutse Khomh, Ying Zou, "Classifying Field Crash Reports for Fixing Bugs: A Case Study of Mozilla Firefox", in Proceedings of the 2011 27th IEEE International Conference on Software Maintenance (ICSM '11), pp. 333-342.

[4] G. Lohman, J. Champlin, and P. Sohn. 2005. Quickly Finding Known Software Problems via Automated Symptom Matching. In Proc. of the Second International Conference on Automatic Computing. 101–110.

[5] Y. Dang, R. Wu, H. Zhang, D. Zhang, and P. Nobel, "ReBucket: a method for clustering duplicate crash reports based on call stack similarity", in Proceedings of the 34th International Conference on Software Engineering (ICSE '12), pp. 1084-1093.

[6] J. Lerch, and M, Mezini, "Finding Duplicates of Your Yet Unwritten Bug Report", in Proceedings of the 2013 17th European Conference on Software Maintenance and Reengineering (CSMR '13). IEEE Computer Society, Washington, DC, USA, 69-78.

[7] J. C. Campbell, E. A. Santos, and A. Hindle, "The unreasonable effectiveness of traditional information retrieval in crash report deduplication", in Proceedings of the 13th International Conference on Mining Software Repositories (MSR '16). ACM, New York, NY, USA, 269-280.

[8] S.D. Bay, and M.J. Pazzani, "Detecting change in categorical data: Mining contrast sets", in Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'99), pp. 302–306, San Diego, U.S.A., August 1999.

[9] S.D. Bay, and M.J. Pazzani, "Detecting group differences: Mining contrast sets", Data Mining and Knowledge Discovery, 5(3):213–246, 2001.

[10] R. J. Hilderman, and T. Peckham, "A Statistically Sound Alternative Approach to Mining Contrast Sets", in Proceedings of the 4th Australasian Data Mining Conference (AusDM), 2005.

[11] https://mozilla.github.io/process-releases/draft/development_overview/.

[12] M. V. Mäntylä, B. Adams, F. Khomh, E. Engström, and K. Petersen, "On rapid releases and software testing: a case study and a semi-systematic literature review", Empirical Softw. Engg. 20, 5 (October 2015), 1384-1425.

[13] J. Śliwerski, T. Zimmermann, and A. Zeller, "When do changes induce fixes?", in Proceedings of the 2005 international workshop on Mining software repositories (MSR '05). ACM, New York, NY, USA, 1-5.

[14] Mozilla, "Bugzilla - https://bugzilla.mozilla.org/".

[15] Mozilla, "Web compatibility issue tracker - https://webcompat.com/".

[16] Kunihiko Fukushima, “Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position,” Biological Cybernetics, vol. 36, no. 4, pp. 193–202, Apr 1980.

[17] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, “Backpropagation applied to handwritten zip code recognition,” Neural Comput., vol. 1, no. 4, pp. 541–551, Dec. 1989.

[18] Raia Hadsell, Sumit Chopra, and Yann LeCun, “Dimensionality reduction by learning an invariant mapping,” in Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2, Washington, DC, USA, 2006, CVPR ’06, pp. 1735–1742, IEEE Computer Society.

[19] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A Large-Scale Hierarchical Image Database,” in CVPR09, 2009.

[20] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” CoRR, vol. abs/1409.1556, 2014.

4. Products

I Year

M. Castelluccio, G. Poggi, C. Sansone, L. Verdoliva – Land Use Classification in Remote Sensing Images by Convolutional Neural Networks – https://arxiv.org/abs/1508.00092 – 93 citations

II Year

M. Castelluccio, G. Poggi, C. Sansone, L. Verdoliva – Training Convolutional Neural Networks for Semantic Classification of Remote Sensing Imagery – Joint Urban Remote Sensing Event (JURSE2017)

M. Castelluccio, C. Sansone, L. Verdoliva, and G. Poggi – Automatically analyzing groups of crashes for finding correlations – Proceedings of the 2017 11th Joint Meeting of the

European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering (ESEC/FSE 2017)

M. Castelluccio, L. An, and F. Khomh – Is It Safe to Uplift This Patch? An Empirical Study on Mozilla Firefox – In proceedings of the 33rd International Conference on Software Maintenance and Evolution (ICSME 2017). Received IEEE TCSE Distinguished Paper Award. Invited for publication on “Empirical Software Engineering” journal.

5. Conferences

Presentation of the paper ‘Automatically analyzing groups of crashes for finding correlations’ at the 11th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering, PADERBORN, GERMANY, September 04 – 08.